

Micron StrataFlash Embedded Memory

**MT28GU256AAA2EGC-0AAT, MT28GU512AAA2EGC-0AAT,
MT28GU01GAAA2EGC-0AAT**

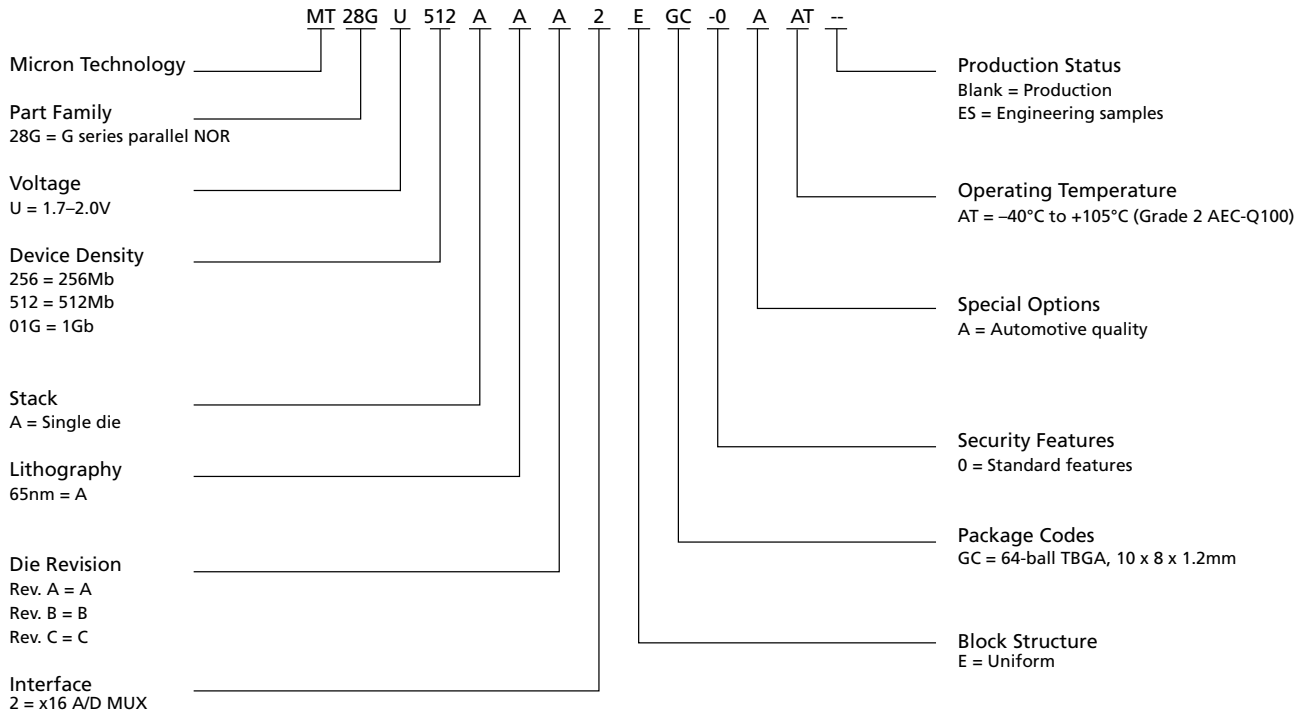
Features

- High-performance read, program, and erase
 - 106ns initial read access
 - 108 MHz with zero wait-state synchronous burst reads: 7ns clock-to-data output
 - 133 MHz with zero wait-state synchronous burst reads: 6ns clock-to-data output
 - 8-, 16-, and continuous-word synchronous-burst reads
 - Programmable WAIT configuration
 - Customer-configurable output driver impedance
 - Buffered programming: 2.0 μ s/word (TYP), 512Mb
 - Block erase: 0.9s per block (TYP)
 - 20 μ s (TYP) program/erase suspend
- Architecture
 - 16-bit wide data bus
 - Multilevel cell technology
 - Symmetrically-blocked array architecture
 - 256KB erase blocks
 - 1Gb device: Eight 128Mb partitions
 - 512Mb device: Eight 64Mb partitions
 - 256Mb device: Eight 32Mb partitions
 - Status register for partition/device status
 - Blank check feature
- Quality and reliability
 - Automotive temperature: -40°C to $+105^{\circ}\text{C}$ (Grade 2 AEC-Q100)
 - Minimum 100,000 ERASE cycles per block
 - More than 20 years data retention
 - 65nm process technology
- Power
 - Core voltage: 1.7– 2.0V
 - I/O voltage: 1.7–2.0V
 - Standby current: 60 μ A (TYP) for 512Mb
 - Automatic power savings mode
 - 16-word synchronous-burst read current: 23mA (TYP) @ 108 MHz; 24mA (TYP) @ 133 MHz
- Software
 - Micron[®] Flash data integrator (FDI) optimized
 - Basic command set (BCS) and extended command set (ECS) compatible
 - Common Flash interface (CFI) capable
- Security
 - One-time programmable (OTP) space
 - 64 unique factory device identifier bits
 - 2112 user-programmable OTP bits
 - Absolute write protection: $V_{\text{PP}} = \text{GND}$
 - Power-transition erase/program lockout
 - Individual zero latency block locking
 - Individual block lock-down
- Density and packaging
 - 256Mb, 512Mb, and 1Gb
 - Address-data multiplexed interface
 - 64-Ball TBGA

Part Numbering Information

Devices are shipped from the factory with memory content bits erased to 1. For available options, such as packages or high/low protection, or for further information, contact the factory. Available part numbers can be verified at www.micron.com. Feature and specification comparison by device type is available at www.micron.com/products. Contact the factory for devices not found.

Figure 1: Current Part Number Decoder



Contents

| | |
|---|----|
| Important Notes and Warnings | 8 |
| General Description | 8 |
| Functional Overview | 9 |
| Configuration and Memory Map | 10 |
| Device ID | 13 |
| Package Dimensions | 14 |
| Signal Assignments | 15 |
| Signal Descriptions | 16 |
| Bus Interface | 18 |
| Reset | 18 |
| Standby | 18 |
| Output Disable | 18 |
| Asynchronous Read | 19 |
| Synchronous Read | 19 |
| Burst Wrapping | 19 |
| End-of-Wordline Delay | 20 |
| Write | 21 |
| Command Definitions | 22 |
| Status Register | 24 |
| Clear Status Register | 25 |
| Read Configuration Register | 26 |
| Programming the Read Configuration Register | 27 |
| Latency Count Code and Clock Frequency | 28 |
| Extended Configuration Register | 29 |
| Output Driver Control | 29 |
| Programming the Extended Configuration Register | 30 |
| Read Operations | 31 |
| Read Array | 31 |
| Read ID | 31 |
| Read CFI | 32 |
| Read Status Register | 32 |
| WAIT Operation | 33 |
| Programming Modes | 34 |
| Control Mode | 34 |
| Object Mode | 35 |
| Program Operations | 39 |
| Single-Word Programming | 39 |
| Buffered Programming | 40 |
| Buffered Enhanced Factory Programming | 40 |
| Erase Operations | 43 |
| BLOCK ERASE | 43 |
| SUSPEND and RESUME Operations | 44 |
| SUSPEND Operation | 44 |
| RESUME Operation | 45 |
| BLANK CHECK Operation | 46 |
| Block Lock | 47 |
| One-Time Programmable Operations | 49 |
| Programming OTP Area | 51 |
| Reading OTP Area | 51 |
| Global Main-Array Protection | 52 |



| | |
|---|-----|
| Power and Reset Specifications | 53 |
| Initialization | 53 |
| Power-Up and Down | 53 |
| Reset | 53 |
| Automatic Power Saving | 55 |
| Power Supply Decoupling | 55 |
| Electrical Specifications | 56 |
| Electrical Specifications – DC Current and Voltage Characteristics and Operating Conditions | 57 |
| Electrical Specifications – AC Characteristics and Operating Conditions | 60 |
| AC Test Conditions | 60 |
| AC Read Specifications | 62 |
| AC Read Timing | 63 |
| AC Write Specifications | 68 |
| Electrical Specifications – Program/Erase Characteristics | 73 |
| Common Flash Interface | 74 |
| READ CFI Structure Output | 74 |
| CFI ID String | 75 |
| System Interface Information | 75 |
| Device Geometry Definition | 76 |
| Primary Micron-Specific Extended Query | 78 |
| Program Flowcharts | 84 |
| Erase Flowcharts | 92 |
| Block Lock and Protection Flowcharts | 96 |
| Blank Check Flowcharts | 100 |
| AADM Mode | 103 |
| AADM Feature Overview | 103 |
| AADM Mode Enable (RCR[4] = 1) | 103 |
| Bus Cycles and Address Capture | 103 |
| WAIT Behavior | 103 |
| Asynchronous READ and WRITE Cycles | 104 |
| Asynchronous READ Cycles | 104 |
| Asynchronous WRITE Cycles | 106 |
| Synchronous READ and WRITE Cycles | 107 |
| Synchronous READ Cycles | 107 |
| Synchronous WRITE Cycles | 110 |
| System Boot | 110 |
| Revision History | 111 |
| Rev. F – 5/18 | 111 |
| Rev. E – 6/15 | 111 |
| Rev. D – 4/15 | 111 |
| Rev. C – 2/15 | 111 |
| Rev. B – 10/14 | 111 |
| Rev. A – 12/13 | 111 |

List of Figures

| | |
|--|-----|
| Figure 1: Current Part Number Decoder | 2 |
| Figure 2: 64-Ball TBGA (10mm x 8mm x 1.2mm) – Package Code: GC | 14 |
| Figure 3: 64-Ball TBGA (Top View, Balls Down) | 15 |
| Figure 4: Main Array Word Lines | 20 |
| Figure 5: Wrap/No-Wrap Example | 20 |
| Figure 6: End-of-Wordline Delay | 20 |
| Figure 7: Configurable Programming Regions: Control Mode and Object Mode | 35 |
| Figure 8: Configurable Programming Regions: Control Mode and Object Mode Segments | 37 |
| Figure 9: BLOCK LOCK Operations | 48 |
| Figure 10: OTP Area Map | 50 |
| Figure 11: V _{PP} Supply Connection Example | 52 |
| Figure 12: RESET Operation Waveforms | 54 |
| Figure 13: AC Input/Output Reference Waveform | 60 |
| Figure 14: Transient Equivalent Testing Load Circuit | 60 |
| Figure 15: Clock Input AC Waveform | 61 |
| Figure 16: Asynchronous Single-Word Read | 64 |
| Figure 17: Synchronous 8- or 16-Word Burst Read (A/D MUX) | 65 |
| Figure 18: Synchronous Continuous Misaligned Burst Read (A/D MUX) | 66 |
| Figure 19: Synchronous Burst with Burst-Interrupt (AD-MUX) | 66 |
| Figure 20: Write Timing | 69 |
| Figure 21: Write to Write (A/D-MUX) | 70 |
| Figure 22: Async Read to Write (A/D-MUX) | 70 |
| Figure 23: Write to Async Read (A/D-MUX) | 71 |
| Figure 24: Sync Read to Write (A/D-MUX) | 71 |
| Figure 25: Write to Sync Read (A/D-MUX) | 72 |
| Figure 26: Word Program Flow Chart | 84 |
| Figure 27: Word Program Full Status Check Flow Chart | 85 |
| Figure 28: Program Suspend/Resume Flow Chart | 86 |
| Figure 29: Buffer Programming Flow Chart | 88 |
| Figure 30: Buffered Enhanced Factory Programming (BEFP) Flow Chart | 90 |
| Figure 31: Block Erase Flowchart | 92 |
| Figure 32: Block Erase Full Status Check Flow Chart | 93 |
| Figure 33: Erase Suspend/Resume Flow Chart | 94 |
| Figure 34: Block Lock Operations Flow Chart | 96 |
| Figure 35: Protection Register Programming Flow Chart | 97 |
| Figure 36: Protection Register Programming Full Status Check Flow Chart | 99 |
| Figure 37: Blank Check Flow Chart | 100 |
| Figure 38: Blank Check Full Status Check Flow Chart | 102 |
| Figure 39: AADM Asynchronous READ Cycle (Latching A[MAX:0]) | 105 |
| Figure 40: AADM Asynchronous READ Cycle (Latching A[15:0] only) | 105 |
| Figure 41: AADM Asynchronous WRITE Cycle (Latching A[MAX:0]) | 106 |
| Figure 42: AADM Asynchronous WRITE Cycle (Latching A[15:0] only) | 107 |
| Figure 43: AADM Synchronous Burst READ Cycle (ADV# De-asserted Between Address Cycles) | 109 |
| Figure 44: AADM Synchronous Burst READ Cycle (ADV# Not De-asserted Between Address Cycles) | 109 |
| Figure 45: AADM Synchronous Burst READ Cycle (Latching A[15:0] only) | 110 |

List of Tables

| | |
|---|----|
| Table 1: Main Array Memory Map – 256Mb | 10 |
| Table 2: Main Array Memory Map – 512Mb, 1Gb | 11 |
| Table 3: Device ID Codes | 13 |
| Table 4: Signal Descriptions | 16 |
| Table 5: Address Mapping for Address/Data MUX Interface | 17 |
| Table 6: Bus Control Signals | 18 |
| Table 7: Command Set | 22 |
| Table 8: Status Register Bit Definitions (Default Value = 0080h) | 24 |
| Table 9: CLEAR STATUS REGISTER Command Bus Cycles | 25 |
| Table 10: Read Configuration Register Bit Definitions | 26 |
| Table 11: PROGRAM READ CONFIGURATION REGISTER Bus Cycles | 27 |
| Table 12: Supported Latency and Clock Frequency | 28 |
| Table 13: Extended Configuration Register Bit Definitions (Default Value = 0004h) | 29 |
| Table 14: Output Driver Control Characteristics | 29 |
| Table 15: Program Extended Configuration Register Command Bus Cycles | 30 |
| Table 16: READ MODE Command Bus Cycles | 31 |
| Table 17: Device Information | 32 |
| Table 18: WAIT Behavior Summary – A/D MUX | 33 |
| Table 19: Programming Region Next State | 38 |
| Table 20: PROGRAM Command Bus Cycles | 39 |
| Table 21: BEFP Requirements and Considerations | 41 |
| Table 22: ERASE Command Bus Cycle | 43 |
| Table 23: Valid Commands During Suspend | 44 |
| Table 24: SUSPEND and RESUME Command Bus Cycles | 45 |
| Table 25: BLANK CHECK Command Bus Cycles | 46 |
| Table 26: BLOCK LOCK Command Bus Cycles | 47 |
| Table 27: Block Lock Configuration | 48 |
| Table 28: Program OTP Area Command Bus Cycles | 49 |
| Table 29: Power Sequencing | 53 |
| Table 30: Reset Specifications | 54 |
| Table 31: Absolute Maximum Ratings | 56 |
| Table 32: Operating Conditions | 56 |
| Table 33: DC Current Characteristics and Operating Conditions ($V_{CCQ} = 1.7V-2.0V$) | 57 |
| Table 34: DC Voltage Characteristics and Operating Conditions ($V_{CCQ} = 1.7V-2.0V$) | 59 |
| Table 35: AC Input Requirements | 60 |
| Table 36: Test Configuration Load Capacitor Values for Worst Case Speed Conditions | 60 |
| Table 37: Capacitance | 61 |
| Table 38: AC Read Specifications (CLK-Latching, 133 MHz), $V_{CCQ} = 1.7V$ to $2.0V$ | 62 |
| Table 39: AC Write Specifications | 68 |
| Table 40: Program/Erase Characteristics | 73 |
| Table 41: Example of CFI Output (x16 Device) as a Function of Device and Mode | 74 |
| Table 42: CFI Database: Addresses and Sections | 74 |
| Table 43: CFI ID String | 75 |
| Table 44: System Interface Information | 75 |
| Table 45: Device Geometry | 76 |
| Table 46: Block Region Map Information | 77 |
| Table 47: Primary Micron-Specific Extended Query | 78 |
| Table 48: One Time Programmable (OTP) Space Information | 79 |
| Table 49: Burst Read Informaton | 80 |
| Table 50: Partition and Block Erase Region Information | 80 |



| | |
|---|-----|
| Table 51: Partition Region 1 Information: Top and Bottom Offset/Address | 81 |
| Table 52: Partition and Erase Block Map Information | 83 |
| Table 53: Word Program Procedure | 84 |
| Table 54: Word Program Full Status Check Procedure | 85 |
| Table 55: Program Suspend/Resume Procedure | 87 |
| Table 56: Buffer Programming Procedure | 89 |
| Table 57: Buffered Enhanced Factory Programming (BEFP) Procedure | 90 |
| Table 58: Block Erase Procedure | 92 |
| Table 59: Block Erase Full Status Check Procedure | 93 |
| Table 60: Erase Suspend/Resume Procedure | 95 |
| Table 61: Block Lock Operations Procedure | 96 |
| Table 62: Protection Register Programming Procedure | 98 |
| Table 63: Protection Register Programming Full Status Check Procedure | 99 |
| Table 64: Blank Check Procedure | 100 |
| Table 65: Blank Check Full Status Check Procedure | 102 |
| Table 66: AADM Asynchronous and Latching Timings | 104 |
| Table 67: AADM Asynchronous Write Timings | 106 |
| Table 68: AADM Synchronous Timings | 107 |

Important Notes and Warnings

Micron Technology, Inc. ("Micron") reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions. This document supersedes and replaces all information supplied prior to the publication hereof. You may not rely on any information set forth in this document if you obtain the product described herein from any unauthorized distributor or other source not authorized by Micron.

Automotive Applications. Products are not designed or intended for use in automotive applications unless specifically designated by Micron as automotive-grade by their respective data sheets. Distributor and customer/distributor shall assume the sole risk and liability for and shall indemnify and hold Micron harmless against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, death, or property damage resulting directly or indirectly from any use of non-automotive-grade products in automotive applications. Customer/distributor shall ensure that the terms and conditions of sale between customer/distributor and any customer of distributor/customer (1) state that Micron products are not designed or intended for use in automotive applications unless specifically designated by Micron as automotive-grade by their respective data sheets and (2) require such customer of distributor/customer to indemnify and hold Micron harmless against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, death, or property damage resulting from any use of non-automotive-grade products in automotive applications.

Critical Applications. Products are not authorized for use in applications in which failure of the Micron component could result, directly or indirectly in death, personal injury, or severe property or environmental damage ("Critical Applications"). Customer must protect against death, personal injury, and severe property and environmental damage by incorporating safety design measures into customer's applications to ensure that failure of the Micron component will not result in such harms. Should customer or distributor purchase, use, or sell any Micron component for any critical application, customer and distributor shall indemnify and hold harmless Micron and its subsidiaries, subcontractors, and affiliates and the directors, officers, and employees of each against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such critical application, whether or not Micron or its subsidiaries, subcontractors, or affiliates were negligent in the design, manufacture, or warning of the Micron product.

Customer Responsibility. Customers are responsible for the design, manufacture, and operation of their systems, applications, and products using Micron products. ALL SEMICONDUCTOR PRODUCTS HAVE INHERENT FAILURE RATES AND LIMITED USEFUL LIVES. IT IS THE CUSTOMER'S SOLE RESPONSIBILITY TO DETERMINE WHETHER THE MICRON PRODUCT IS SUITABLE AND FIT FOR THE CUSTOMER'S SYSTEM, APPLICATION, OR PRODUCT. Customers must ensure that adequate design, manufacturing, and operating safeguards are included in customer's applications and products to eliminate the risk that personal injury, death, or severe property or environmental damages will result from failure of any semiconductor component.

Limited Warranty. In no event shall Micron be liable for any indirect, incidental, punitive, special or consequential damages (including without limitation lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort, warranty, breach of contract or other legal theory, unless explicitly stated in a written agreement executed by Micron's duly authorized representative.

General Description

Micron's 65nm device is the latest generation of StrataFlash[®] memory. The device provides high-performance, asynchronous read mode and synchronous-burst read mode using 1.8V low-voltage, multilevel cell (MLC) technology.

The device is manufactured using 65nm process technologies and is available in industry-standard chip scale packaging.

Functional Overview

This device provides high read and write performance at low voltage on a 16-bit data bus. The architecture provides individually erasable memory blocks sized for optimum code and data storage.

The device supports synchronous burst reads up to 133 MHz using CLK latching.

Upon initial power-up or return from reset, the device defaults to asynchronous read mode. Configuring the read configuration register enables synchronous burst mode reads. In synchronous burst mode, output data is synchronized with a user-supplied clock signal. In continuous-burst mode, a data read can traverse partition boundaries. A WAIT signal simplifies synchronizing the CPU to the memory.

Designed for low-voltage applications, the device supports READ operations with V_{CC} at 1.8V, and ERASE and PROGRAM operations with V_{PP} at 1.8V or 9.0V. V_{CC} and V_{PP} can be tied together for a simple, ultra low-power design. In addition to voltage flexibility, a dedicated V_{PP} connection provides complete data protection when V_{PP} is less than V_{PPLK} .

A status register provides status and error conditions of ERASE and PROGRAM operations.

One-time programmable (OTP) area enables unique identification that can be used to increase security. Additionally, the individual block lock feature provides zero-latency block locking and unlocking to protect against unwanted program or erase of the array.

The device offers power-savings features, including automatic power savings mode and standby mode. For power savings, the device automatically enters APS following a READ cycle. Standby is initiated when the system deselects the device by de-asserting CE#.



Configuration and Memory Map

The device features a symmetrical block architecture.

The main array of the 256Mb device is divided into eight 32Mb partitions. Each partition is divided into sixteen 256KB blocks (8 x 16 = 128 blocks).

The main array of the 512Mb device is divided into eight 64Mb partitions. Each partition is divided into thirty-two 256KB blocks (8 x 32 = 256 blocks).

The main array of the 1Gb device is divided into eight 128Mb partitions. Each partition is divided into sixty-four 256KB blocks (8 x 64 = 512 blocks).

Each block is divided into as many as 256 1KB programming regions. Each region is divided into as many as thirty-two 32-byte segments

Table 1: Main Array Memory Map – 256Mb

| Partition | Size (Mb) | Block # | Address Range |
|-----------|-----------|---------|-------------------|
| 7 | 32 | 127 | 0FE0000-0FFFFFFF |
| | | . | . |
| | | . | . |
| 6 | 32 | 112 | 0E00000-0E1FFFFF |
| | | . | . |
| | | . | . |
| 5 | 32 | 96 | 0C00000-0C1FFFFF |
| | | . | . |
| | | . | . |
| 4 | 32 | 80 | 0A00000-0A1FFFFF |
| | | . | . |
| | | . | . |
| 3 | 32 | 79 | 09E0000-09FFFFFFF |
| | | . | . |
| | | . | . |
| | | 64 | 0800000-081FFFFF |
| | | . | . |
| | | . | . |
| | | 63 | 07E0000-07FFFFFFF |
| | | . | . |
| | | . | . |
| | | 48 | 0600000-061FFFFF |
| | | . | . |
| | | . | . |



Table 1: Main Array Memory Map – 256Mb (Continued)

| Partition | Size (Mb) | Block # | Address Range |
|-----------|-----------|---------|------------------|
| 2 | 32 | 47 | 05E0000-05FFFFFF |
| | | . | . |
| | | . | . |
| 1 | 32 | 32 | 0400000-041FFFF |
| | | . | . |
| | | . | . |
| 0 | 32 | 16 | 03E0000-03FFFFFF |
| | | . | . |
| | | . | . |
| 0 | 32 | 15 | 0200000-021FFFF |
| | | . | . |
| | | . | . |
| 0 | 32 | 0 | 01E0000-01FFFFFF |
| | | . | . |
| | | . | . |
| 0 | 32 | 0 | 0000000-001FFFF |
| | | . | . |
| | | . | . |

Table 2: Main Array Memory Map – 512Mb, 1Gb

| Partition | 512Mb | | | 1Gb | | |
|-----------|-----------|---------|------------------|-----------|---------|------------------|
| | Size (Mb) | Block # | Address Range | Size (Mb) | Block # | Address Range |
| 7 | 64 | 255 | 1FE0000-1FFFFFFF | 128 | 511 | 3FE0000-3FFFFFFF |
| | | . | . | | . | . |
| | | . | . | | . | . |
| 6 | 64 | 224 | 1C00000-1C1FFFF | 128 | 448 | 3800000-381FFFF |
| | | . | . | | . | . |
| | | . | . | | . | . |
| 5 | 64 | 192 | 1800000-181FFFF | 128 | 447 | 37E0000-37FFFFFF |
| | | . | . | | . | . |
| | | . | . | | . | . |
| 5 | 64 | 160 | 1400000-141FFFF | 128 | 384 | 3000000-301FFFF |
| | | . | . | | . | . |
| | | . | . | | . | . |
| 5 | 64 | 191 | 17E0000-17FFFFFF | 128 | 383 | 2FE0000-2FFFFFFF |
| | | . | . | | . | . |
| | | . | . | | . | . |



Table 2: Main Array Memory Map – 512Mb, 1Gb (Continued)

| Partition | 512Mb | | | 1Gb | | |
|-----------|-----------|---------|------------------|-----------|---------|------------------|
| | Size (Mb) | Block # | Address Range | Size (Mb) | Block # | Address Range |
| 4 | 64 | 159 | 13E0000-13FFFFFF | 128 | 319 | 27E0000-27FFFFFF |
| | | . | . | | . | . |
| | | 128 | 1000000-101FFFFF | | 256 | 2000000-201FFFFF |
| 3 | 64 | 127 | 0FE0000-0FFFFFFF | 128 | 255 | 1FE0000-1FFFFFFF |
| | | . | . | | . | . |
| | | 96 | 0300000-031FFFFF | | 192 | 1800000-181FFFFF |
| 2 | 64 | 95 | 0BE0000-0BFFFFFF | 128 | 191 | 17E0000-17FFFFFF |
| | | . | . | | . | . |
| | | 64 | 0800000-081FFFFF | | 128 | 1000000-101FFFFF |
| 1 | 64 | 63 | 07E0000-07FFFFFF | 128 | 127 | 0FE0000-0FFFFFFF |
| | | . | . | | . | . |
| | | 32 | 0400000-041FFFFF | | 64 | 0800000-081FFFFF |
| 0 | 64 | 31 | 03E0000-03FFFFFF | 128 | 63 | 07E0000-07FFFFFF |
| | | . | . | | . | . |
| | | 0 | 0000000-001FFFFF | | 0 | 0000000-001FFFFF |



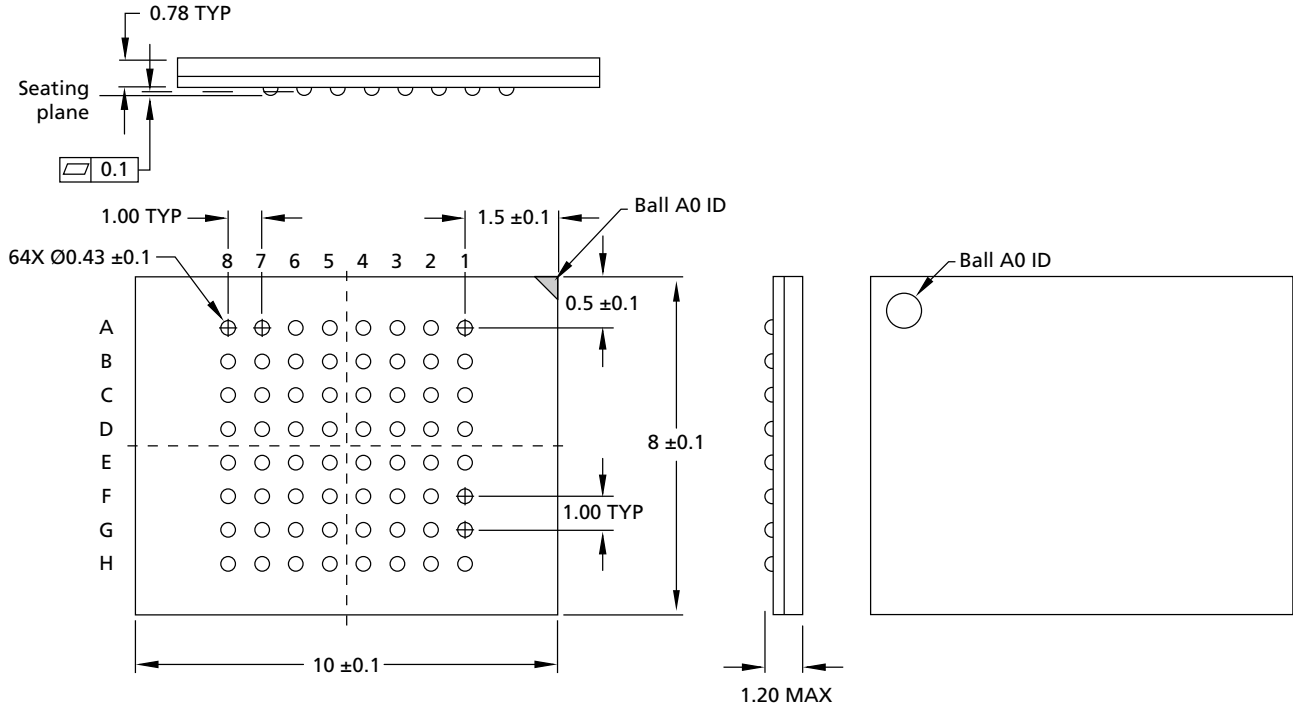
Device ID

Table 3: Device ID Codes

| Density | Product | Device Identifier Code (Hex) |
|---------|---------|------------------------------|
| 256Mb | A/D MUX | 8904 |
| 512Mb | A/D MUX | 8881 |
| 1024Mb | A/D MUX | 88B1 |

Package Dimensions

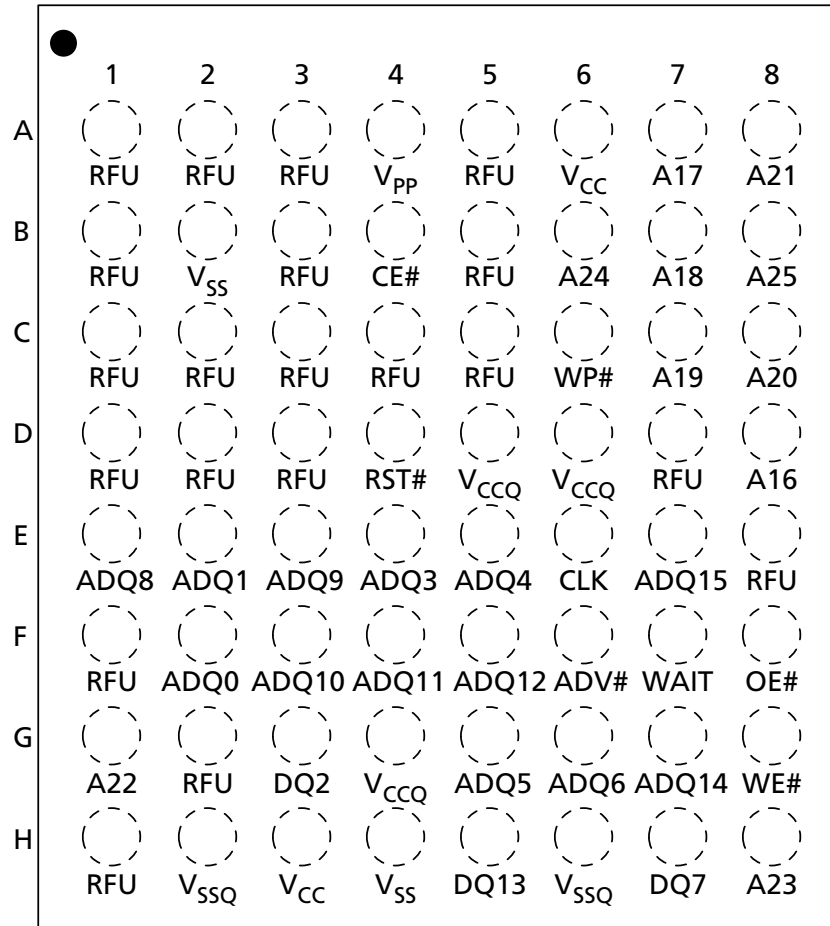
Figure 2: 64-Ball TBGA (10mm x 8mm x 1.2mm) – Package Code: GC



Note: 1. All dimensions are in millimeters.

Signal Assignments

Figure 3: 64-Ball TBGA (Top View, Balls Down)



- Notes:
1. B6 is A24 for 512Mb densities and above; otherwise, it is a no connect (NC).
 2. B8 is A25 for 1Gb density; otherwise, it is a no connect (NC).
 3. For AA/D MUX configuration, the upper addresses A[MAX;16] must be connected to V_{SS}.

Signal Descriptions

Table 4: Signal Descriptions

| Symbol | Type | Description |
|------------------------|--------------|---|
| A/D MUX | | |
| A[MAX:16] | Input | Address inputs: Upper address inputs for all READ/WRITE cycles. |
| A/DQ[15:0] | Input/Output | Address inputs or data: Lower address inputs during the address phase for all READ/WRITE cycles; data or command inputs during WRITE cycles; data, status, or device information outputs during READ cycles. |
| Control Signals | | |
| CE# | Input | Chip enable: LOW true input. When LOW, CE# selects the die; when HIGH, CE# deselects the die and places it in standby. |
| OE# | Input | Output enable: LOW true input. Must be LOW for READs and HIGH for WRITEs. |
| WE# | Input | Write enable: LOW true input. Must be LOW for WRITEs and HIGH for READs. |
| CLK | Input | Clock: Synchronizes burst READ operations with the host controller. |
| ADV# | Input | Address valid: LOW true input. When LOW, ADV# enables address inputs. For synchronous burst READs, address inputs are latched on the rising edge. |
| WP# | Input | Write protect: LOW true input. When LOW, WP# enables block lock down; when HIGH, WP# disables block lock down. |
| RST# | Input | Reset: LOW true input. When LOW, RST# inhibits all operations; must be HIGH for normal operations. |
| V _{PP} | Input | Erase/program voltage: Enables voltage for PROGRAM and ERASE operations. Array contents cannot be altered when V _{PP} is at or below V _{PPLK} . |
| WAIT | Output | WAIT: Configurable HIGH or LOW true output. When asserted, WAIT indicates DQ[15:0] is invalid; when de-asserted, WAIT indicates DQ[15:0] is valid. |
| V _{CC} | Power | Core power: Supply voltage for core circuits. All operations are inhibited when V _{CC} is at or below V _{LKO} . |
| V _{CCQ} | Power | I/O power: Supply voltage for all I/O drivers. All operations are inhibited when V _{CCQ} is at or below V _{LKQ} . |
| V _{SS} | Power | Logic ground: Core logic ground return. Connect all V _{SS} balls to system ground; do not float any V _{SS} balls. |
| V _{SSQ} | Power | I/O ground: I/O driver ground return. Connect all V _{SSQ} balls to system ground; do not float any V _{SSQ} balls. |
| RFU | Reserved | Reserved: Reserved for future use and should not be connected. |

Table 5: Address Mapping for Address/Data MUX Interface

| Address Bit | A/D MUX Configuration (RCR Bit 4 = 0) and OE# = 1 | AADM Mode (RCR Bit 4 = 1) and OE# = 1 | AADM Mode (RCR Bit 4 = 1) and OE# = 0 |
|-------------|--|--|--|
| A0 | DQ0 | A0 | A16 |
| A1 | DQ1 | A1 | A17 |
| A2 | DQ2 | A2 | A18 |
| A3 | DQ3 | A3 | A19 |
| A4 | DQ4 | A4 | A20 |
| A5 | DQ5 | A5 | A21 |
| A6 | DQ6 | A6 | A22 |
| A7 | DQ7 | A7 | A23 |
| A8 | DQ8 | A8 | A24 |
| A9 | DQ9 | A9 | A25 |
| A10 | DQ10 | A10 | – |
| A11 | DQ11 | A11 | – |
| A12 | DQ12 | A12 | – |
| A13 | DQ13 | A13 | – |
| A14 | DQ14 | A14 | – |
| A15 | DQ15 | A15 | – |
| A16 | A16 | – | – |
| A17 | A17 | – | – |
| A18 | A18 | – | – |
| A19 | A19 | – | – |
| A20 | A20 | – | – |
| A21 | A21 | – | – |
| A22 | A22 | – | – |
| A23 | A23 | – | – |
| A24 | A24 | – | – |
| A25 | A25 | – | – |

Bus Interface

The bus interface uses CMOS-compatible address, data, and bus control signals for all bus WRITE and bus READ operations. The address signals are input only, the data signals are input/output (I/O), and the bus control signals are input only. The address inputs are used to specify the internal device location during bus READ and bus WRITE operations. The data I/Os carry commands, data, or status to and from the device. The control signals are used to select and deselect the device, indicate a bus READ or bus WRITE operation, synchronize operations, and reset the device.

Do not float any inputs. All inputs must be driven or terminated for proper device operation. Some features may use additional signals. See Signal Descriptions for descriptions of these signals.

The following table shows the logic levels that must be applied to the bus control signal inputs for the bus operations listed.

Table 6: Bus Control Signals

X = Don't Care; High = V_{IH} ; Low = V_{IL}

| Bus Operations | RST# | CE# | CLK | ADV# | OE# | WE# | Address | Data I/O |
|-------------------|------|------|---------|--------|------|------|---------|----------|
| RESET | LOW | X | X | X | X | X | X | High-Z |
| STANDBY | HIGH | HIGH | X | X | X | X | X | High-Z |
| OUTPUT DISABLE | HIGH | X | X | X | HIGH | X | X | High-Z |
| Asynchronous READ | HIGH | LOW | X | LOW | LOW | HIGH | Valid | Output |
| Synchronous READ | HIGH | LOW | Running | Toggle | LOW | HIGH | Valid | Output |
| WRITE | HIGH | LOW | X | X | HIGH | LOW | Valid | Input |

Reset

RST# LOW places the device in reset, where device operations are disabled; inputs are ignored, and outputs are placed in High-Z.

Any ongoing ERASE or PROGRAM operation will be aborted and data at that location will be indeterminate.

RST# HIGH enables normal device operations. A minimum delay is required before the device is able to perform a bus READ or bus WRITE operation. See AC specifications.

Standby

RST# HIGH and CE# HIGH place the device in standby, where all other inputs are ignored, outputs are placed in High-Z (independent of the level placed on OE#), and power consumption is substantially reduced.

Any ongoing ERASE or PROGRAM operation continues in the background and the device draws active current until the operation has finished.

Output Disable

When OE# is de-asserted with CE# asserted, the device outputs are disabled. Output pins are placed in High-Z. WAIT is de-asserted.

Asynchronous Read

For RCR15 = 1 (default), CE# LOW and OE# LOW place the device in asynchronous bus read mode:

- RST# and WE# must be held HIGH; CLK must be tied either HIGH or LOW.
- Address inputs must be held stable throughout the access, or latched with ADV#.
- ADV# must be held LOW or can be toggled to latch the address.
- Valid data is output on the data I/Os after t_{AVQV} , t_{ELQV} , t_{VLQV} , or t_{GLQV} , whichever is satisfied last.

Asynchronous READ operations are independent of the voltage level on V_{PP} .

Synchronous Read

For RCR15 = 0, CE# LOW, OE# LOW, and ADV# LOW place the device in synchronous bus read mode:

- RST# and WE# must be held HIGH.
- CLK must be running.
- The first data word is output t_{CHQV} after the latency count has been satisfied.
- For array reads, the next address data is output t_{CHQV} after valid CLK edges until the burst length is satisfied.
- For nonarray reads, the same address data is output t_{CHQV} after valid CLK edges until the burst length is satisfied.

The address for synchronous read operations is latched on the ADV# rising edge or the first rising CLK edge after ADV# LOW, whichever occurs first for devices that support up to 108 MHz. For devices that support up to 133 MHz, the address is latched on the last CLK edge when ADV# is LOW.

Burst Wrapping

Data stored within the memory array is arranged in rows or word lines. During synchronous burst reads, data words are sensed in groups from the array. The starting address of a synchronous burst read determines which word within the wordgroup is output first, and subsequent words are output in sequence until the burst length is satisfied.

The setting of the burst wrap bit (RCR3) determines whether synchronous burst reads will wrap within the wordgroup or continue on to the next wordgroup.

Figure 4: Main Array Word Lines

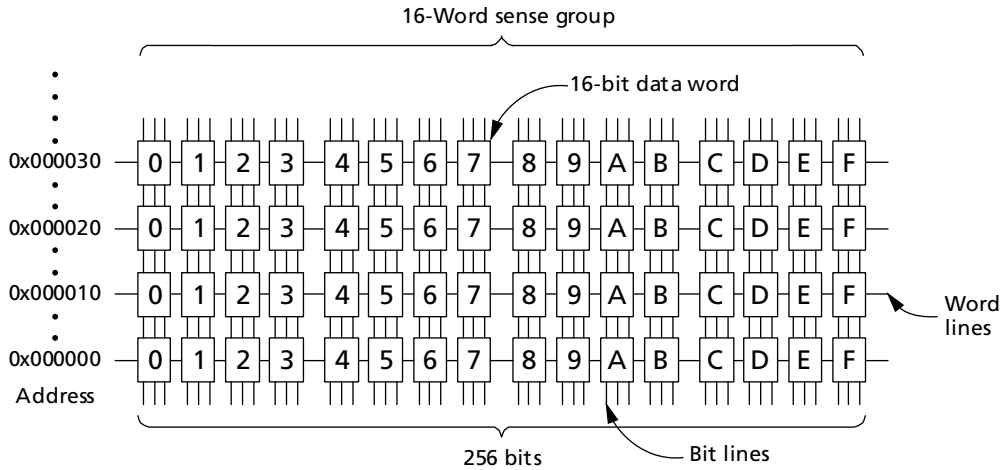
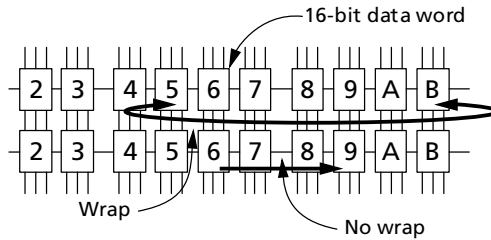


Figure 5: Wrap/No-Wrap Example

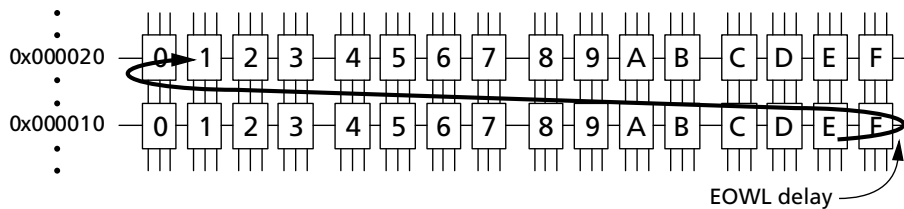


End-of-Wordline Delay

Output delays may occur when the burst sequence crosses the first end-of-wordline boundary onto the start of the next wordline.

No delays occur if the starting address is sense-group aligned or if the burst sequence never crosses a wordline boundary. However, if the starting address is not sense-group aligned, the worst-case end-of-wordline delay is one clock cycle less than the initial access latency count used. This delay occurs only once during the burst access. WAIT informs the system of this delay when it occurs.

Figure 6: End-of-Wordline Delay



Write

CE# LOW and WE# LOW place the device in bus write mode, where RST# and OE# must be HIGH, CLK and ADV# are ignored, input data and address are sampled on the rising edge of WE# or CE#, whichever occurs first.

During a WRITE operation in MUX devices, address is latched during the rising edge of ADV# OR CE# whichever occurs first and data is latched during the rising edge of WE# OR CE# whichever occurs first.

Bus WRITE cycles are asynchronous only.

The following conditions apply when a bus WRITE cycle occurs immediately before, or immediately after, a bus READ cycle:

- When transitioning from a bus READ cycle to a bus WRITE cycle, CE# or ADV# must toggle after OE# goes HIGH.
- When in synchronous read mode (RCR15 = 0; burst clock running), bus WRITE cycle timings ^tVHWL (ADV# HIGH to WE# LOW), ^tCHWL (CLK HIGH to WE# LOW), and ^tWHCH (WE# HIGH to CLK HIGH) must be met.
- When transitioning from a bus WRITE cycle to a bus READ cycle, CE# or ADV# must toggle after WE# goes HIGH.

Command Definitions

Commands are written to the device to control all operations. Some commands are two-cycle commands that use a SETUP and a CONFIRM command; other commands are single-cycle commands that use only a SETUP command followed by a data READ cycle or data WRITE cycle. Valid commands and their associated command codes are shown in the table below.

Table 7: Command Set

| Command | Code (Setup/Confirm) | Description |
|---|-------------------------|--|
| Register Operations | | |
| PROGRAM READ CONFIGURATION REGISTER | 0060h/0003h | Programs the read configuration register. The desired read configuration register value is placed on the address bus, and written to the read configuration register when the CONFIRM command is issued. |
| PROGRAM EXTENDED CONFIGURATION REGISTER | 0060h/0004h | Programs the extended configuration register. The desired extended configuration register value is placed on the address bus, and written to the read configuration register when the CONFIRM command is issued. |
| PROGRAM OTP AREA | 00C0h | Programs OTP area and OTP lock registers. The desired register data is written to the addressed register on the next WRITE cycle. |
| CLEAR STATUS REGISTER | 0050h | Clears all error bits in the status register. |
| Read Mode Operations | | |
| READ ARRAY | 00FFh | Places the addressed partition in read array mode. Subsequent reads outputs array data. |
| READ STATUS REGISTER | 0070h | Places the addressed partition in read status mode. Subsequent reads outputs status register data. |
| READ ID | 0090h | Places the addressed partition in read ID mode. Subsequent reads from specified address offsets output unique device information. |
| READ CFI | 0098h | Places the addressed partition in read CFI mode. Subsequent reads from specified address offsets output CFI data. |
| Array Programming Operations | | |
| SINGLE-WORD PROGRAM | 0041h | Programs a single word into the array. Data is written to the array on the next WRITE cycle. The addressed partition automatically switches to read status register mode. |
| BUFFERED PROGRAM | 00E9h/00D0h | Initiates and executes a BUFFERED PROGRAM operation. Additional bus READ/WRITE cycles are required between the and confirm commands to properly perform this operation. The addressed partition automatically switches to read status register mode. |

Table 7: Command Set (Continued)

| Command | Code (Setup/Confirm) | Description |
|-----------------------------------|---------------------------------|--|
| BUFFERED ENHANCED FACTORY PROGRAM | 0080h/00D0h | Initiates and executes a BUFFERED ENHANCED FACTORY PROGRAM operation. Additional bus READ/WRITE cycles are required after the CONFIRM command to properly perform this operation. The addressed partition automatically switches to read status register mode. |
| Block Erase Operations | | |
| BLOCK ERASE | 0020h/00D0h | Erases a single, addressed block. The ERASE operation commences when the CONFIRM command is issued. The addressed partition automatically switches to read status register mode. |
| Security Operations | | |
| Lock Block | 0060h/0001h | Sets the lock bit of the addressed block. |
| Unlock Block | 0060h/00D0h | Clears the lock bit of the addressed block. |
| Lock-Down Block | 0060h/002Fh | Sets the lock-down bit of the addressed block. |
| Other Operations | | |
| SUSPEND | 00B0h | Initiates a suspend of a PROGRAM or BLOCK ERASE operation already in progress when issued to any device address SR[6] = 1 indicates erase suspend SR[2] = 1 indicates program suspend |
| RESUME | 00D0h | Resumes a suspended PROGRAM or BLOCK ERASE operation when issued to any device address. A program suspend nested within an erase suspend is resumed first. |
| BLANK CHECK | 00BCh/00D0h | Performs a blank check of an addressed block. The addressed partition automatically switches to read status register mode. |

Status Register

The status register is a 16-bit, read-only register that indicates device status, region status, and operating errors. Upon power-up or exit from reset, the status register defaults to 0080h (device ready, no errors).

The status register has status bits and error bits. Status bits are set and cleared by the device; error bits are only set by the device. Error bits are cleared using the CLEAR STATUS REGISTER command or by resetting the device.

To read from the status register, first issue the READ STATUS REGISTER command and then read from the device. Note that some commands automatically switch from read mode to read status register mode.

Table 8: Status Register Bit Definitions (Default Value = 0080h)

| Bit | Name | Description |
|-------|--|--|
| 15:10 | Reserved | Reserved for future use; these bits will always be set to zero |
| 9:8 | Partition program error | SR[9]/SR[8] 0 0 = Region program successful 1 0 = Region program error: Attempted write with object data to control mode region 0 1 = Region-program error: Attempted rewrite to object mode region 1 1 = Region-program error: Attempted write using illegal command (SR[4] will also be set along with SR[8,9] for the above error conditions) |
| 7 | Device status | 0 = Device is busy; SR[9,8,6:1] are invalid, SR[0] is valid 1 = Device is ready; SR[9:8], SR[6:1] are valid |
| 6 | Erase suspend | 0 = Erase suspend not in effect 1 = Erase suspend in effect |
| 5:4 | Erase error/blank check error program error (command sequence error) | SR[5]/SR[4] 0 0 = PROGRAM or ERASE operation successful 0 1 = Program error: operation aborted 1 0 = Erase error: Operation aborted; Blank check error: Operation failed 1 1 = Command sequence error: Command aborted |
| 3 | V _{PP} error | 0 = V _{PP} within acceptable limits during program or erase 1 = V _{PP} < V _{PPLK} during program or erase; operation aborted |
| 2 | Program suspend | 0 = Program suspend not in effect 1 = Program suspend in effect |
| 1 | Block lock error | 0 = Block not locked during program or erase; operation successful 1 = Block locked during program or erase; operation aborted |
| 0 | Partition status | SR[7]/SR[0] 0 0 = Active PROGRAM or ERASE operation in addressed partition BEFP: Program or verify complete, or ready for data 0 1 = Active PROGRAM or ERASE operation in other partition BEFP: Program or Verify in progress 1 0 = No active PROGRAM or ERASE operation in any partition BEFP: Operation complete 1 1 = Reserved |

Clear Status Register

The status register has status bits and error bits. Status bits are set and cleared by the device; error bits are only set by the device. Error bits are cleared using the CLEAR STATUS REGISTER command or by resetting the device.

Note: Care should be taken to avoid status register ambiguity. If a command sequence error occurs while in erase suspend, SR[5:4] will be set, indicating a command sequence error. When the ERASE operation is resumed (and finishes), any errors that may have occurred during the ERASE operation will be masked by the command sequence error. To avoid this situation, clear the status register prior to resuming any suspended ERASE operation.

The CLEAR STATUS REGISTER command functions independent of the voltage level on V_{pp} . Issuing the CLEAR STATUS REGISTER command places the addressed partition in read status register mode. Other partitions are not affected.

Table 9: CLEAR STATUS REGISTER Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-----------------------|----------------------------------|-------------------------------|------------------------------------|---------------------------------|
| CLEAR STATUS REGISTER | Device address | 0050h | – | – |

Read Configuration Register

The read configuration register is a volatile, 16-bit read/write register used to select bus read modes and to configure synchronous burst read behavior of the device.

The read configuration register is programmed using the PROGRAM READ CONFIGURATION REGISTER command. To read the read configuration register, issue the READ ID command and then read from offset 0005h.

Upon power-up or exit from reset, the read configuration register defaults to asynchronous mode (RCR15 = 1; all other bits are ignored).

Table 10: Read Configuration Register Bit Definitions

| Bit | Name | Description |
|-------|---------------|---|
| 15 | Read mode | 0 = Synchronous burst mode 1 = Asynchronous mode (default) |
| 14:11 | Latency count | 0 0 1 1 = Code 3 0 1 0 0 = Code 4 0 1 0 1 = Code 5 0 1 1 0 = Code 6 0 1 1 1 = Code 7 1 0 0 0 = Code 8 1 0 0 1 = Code 9 1 0 1 0 = Code 10 1 0 1 1 = Code 11 1 1 0 0 = Code 12 1 1 0 1 = Code 13 1 1 1 0 = Code 14 Other bit settings are reserved; see the table below for supported clock frequencies |
| 10 | WAIT polarity | 0 = WAIT signal is LOW-true 1 = WAIT signal is HIGH-true |
| 9 | Reserved | Write 0 to reserved bits |
| 8 | WAIT delay | 0 = WAIT de-asserted with valid data 1 = WAIT de-asserted one clock cycle before valid data |
| 7:5 | Reserved | Write 0 to reserved bits |
| 4 | Bus interface | 0 = A/D MUX (default) 1 = AA/D MUX |
| 3 | Reserved | Write 0 to reserved bits |
| 2:0 | Burst length | 0 1 0 = 8-word burst, wrap only 0 1 1 = 16-word burst, wrap only 1 1 1 = Continuous-burst: linear, no-wrap only Other bit settings are reserved |

Programming the Read Configuration Register

The read configuration register is programmed by issuing the PROGRAM READ CONFIGURATION REGISTER command. The desired RCR[15:0] settings are placed on A[15:0], while the PROGRAM READ CONFIGURATION REGISTER SETUP command is placed on the data bus. Upon issuing the SETUP command, the read mode of the addressed partition is automatically changed to read status register mode.

Next, the CONFIRM command is placed on the data bus while the desired settings for RCR[15:0] are again placed on A[15:0]. Upon issuing the CONFIRM command, the read mode of the addressed partition is automatically switched to read array mode.

Because the desired read configuration register value is placed on the address bus, any hardware-connection offsets between the host's address outputs and the device's address inputs must be taken into account. For example, if the host's address outputs are aligned to the device's address inputs such that host address bit A1 is connected to address bit A0, the desired register value must be left-shifted by one (for example, 2532h << 4A64h) before programming the read configuration register

Synchronous read accesses cannot occur until both the device and the host are in synchronous read mode. Therefore, the software instructions used to perform read configuration register programming and host chip select configuration must be guaranteed not to fetch from the device (instructions must be in system RAM or locked in cache). This also applies when switching back to asynchronous read mode from synchronous read mode.

Table 11: PROGRAM READ CONFIGURATION REGISTER Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-------------------------------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| PROGRAM READ CONFIGURATION REGISTER | RCR settings | 0060h | RCR settings | 0003h |

Latency Count Code and Clock Frequency

Table 12: Supported Latency and Clock Frequency

| Latency Count Code | Clock Frequency $V_{CCQ} = 1.7V \text{ to } 2.0V$ |
|--------------------|--|
| 3 | $\leq 29.7 \text{ MHz}$ |
| 4 | $\leq 39.6 \text{ MHz}$ |
| 5 | $\leq 49.5 \text{ MHz}$ |
| 6 | $\leq 59.4 \text{ MHz}$ |
| 7 | $\leq 69.3 \text{ MHz}$ |
| 8 | $\leq 79.2 \text{ MHz}$ |
| 9 | $\leq 89.1 \text{ MHz}$ |
| 10 | $\leq 99.0 \text{ MHz}$ |
| 11 | $\leq 108.9 \text{ MHz}$ |
| 12 | $\leq 118.8 \text{ MHz}$ |
| 13 | $\leq 128.7 \text{ MHz}$ |
| 14 | $\leq 133.0 \text{ MHz}$ |

Extended Configuration Register

The extended configuration register is a volatile 16-bit, read/write register used to select output-driver strength of the device.

Upon power-up or exit from reset, the extended configuration register defaults to 0004h.

The extended configuration register is programmed using the PROGRAM EXTENDED CONFIGURATION REGISTER command. To read the extended configuration register, issue the READ ID command to a partition, and read from <partition base address> + 06h.

Table 13: Extended Configuration Register Bit Definitions (Default Value = 0004h)

| Bit | Name | Description |
|------|-----------------------|---|
| 15:3 | Reserved | Write 0 to reserved bits |
| 2:0 | Output driver control | 0 0 1 = Code 1 0 1 0 = Code 2 0 1 1 = Code 3 1 0 0 = Code 4 (default) 1 0 1 = Code 5 1 1 0 = Code 6 Other bit settings are reserved |

Output Driver Control

The output driver control bits of the extended configuration register enable adjustment of the device's output-driver strength for DQ[15:0] and WAIT. Upon power-up or reset, ECR[2:0] defaults to 100b for to an output impedance setting of 30 Ohms. To change the output-driver strength, program ECR[2:0] to the desired setting.

Table 14: Output Driver Control Characteristics

| ECR[2:0] | Driver Impedance (at $V_{CCQ}/2$) | Driver Multiplier | Load (Same Speed) |
|----------|---------------------------------------|-------------------|-------------------|
| 0 0 1 | 90 Ohms | 1/3 | 10pF |
| 0 1 0 | 60 Ohms | 1/2 | 15pF |
| 0 1 1 | 45 Ohms | 2/3 | 20pF |
| 1 0 0 | 30 Ohms | 1 | 30pF |
| 1 0 1 | 20 Ohms | 1–1/2 | 35pF |
| 1 1 0 | 15 Ohms | 2 | 40pF |

Programming the Extended Configuration Register

The extended configuration register is programmed by issuing the PROGRAM EXTENDED CONFIGURATION REGISTER command. The desired ECR[15:0] settings are placed on A[15:0], while the PROGRAM EXTENDED CONFIGURATION REGISTER SETUP command is placed on the data bus. Upon issuing the SETUP command, the read mode of the addressed partition is automatically changed to read status register mode.

Next, the CONFIRM command is placed on the data bus while the desired settings for ECR[15:0] are again placed on A[15:0]. Upon issuing the CONFIRM command, the read mode of the addressed partition is automatically switched to read array mode.

Because the desired ECR value is placed on the address bus, any hardware-connection offsets between the host's address outputs and the device's address inputs must be taken into account.

For example, if the host's address outputs are aligned to the device's address inputs such that host address bit A1 is connected to address bit A0, the desired register value must be left-shifted by one (for example, 2532h << 4A64h) before programming the ECR.

Programming the ECR functions independently of the voltage on V_{pp} .

Table 15: Program Extended Configuration Register Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|---|-------------------------------|----------------------------|---------------------------------|------------------------------|
| PROGRAM EXTENDED CONFIGURATION REGISTER | Register Data | 0060h | Register Data | 0004h |

Read Operations

The following types of data can be read from the device: array data (read array), device information (read ID), CFI data (read CFI), and device status (read status register).

Upon power-up or return from reset, the device defaults to read array mode. To change the read mode, the appropriate command must be issued to the device.

The table below shows the command codes used to configure the device for the desired read mode.

Table 16: READ MODE Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|----------------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| READ ARRAY | Partition address | 00FFh | – | – |
| READ STATUS REGISTER | Partition address | 0070h | – | – |
| READ ID | Partition address | 0090h | – | – |
| READ CFI | Partition address | 0098h | – | – |

Read Array

Upon power-up or exit from reset, the device defaults to read array mode. Issuing the READ ARRAY command places the addressed partition in read array mode and can be issued only to a partition that is not actively programming or erasing. Subsequent READ operations output array data from that partition.

The addressed partition remains in read array mode until a different READ command is issued, a PROGRAM or ERASE operation is performed, or a BLOCK LOCK SETUP command is issued in that partition, in which case the read mode automatically changes to read status.

To change a partition that is actively programming or erasing to read array mode, first issue the SUSPEND command. After the operation has been suspended, issue the READ ARRAY command to the partition. When the PROGRAM or ERASE operation is subsequently resumed, the partition will automatically revert back to read status mode.

The READ ARRAY command functions independently of the voltage level on V_{PP} .

Issuing the READ ARRAY command to a partition that is actively programming or erasing causes subsequent reads from that partition to output invalid data. Valid array data is output only after the PROGRAM or ERASE operation has completed.

Read ID

Issuing the READ ID command places the addressed partition in read ID mode. Subsequent reads output device information such as manufacturer code, device identifier code, block lock status, OTP data, or read configuration register data.

The addressed partition remains in read ID mode until a different READ command is issued, or a PROGRAM or ERASE operation is performed in that partition, in which case the read mode automatically changes to read status.

The READ ID command functions independently of the voltage level on V_{PP} .

Issuing the READ ID command to a partition that is actively programming or erasing changes that partition's read mode to read ID mode. Subsequent reads from that partition will not output device information until the PROGRAM or ERASE operation has completed.

Table 17: Device Information

| Device Information | Address Bus | Data Bus |
|---|--------------------------------------|---|
| Device manufacturer code | Partition base address + 00h | 0089h |
| Device ID code | Partition base address + 01h | Device ID |
| Block lock status | Block base address + 02h | D0 = Lock status D1 = Lock-down status |
| Read configuration register | Partition base address + 05h | Configuration register data |
| Extended configuration register | Partition base address + 06h | Extended configuration register data |
| OTP lock register 0 | Partition base address + 80h | Lock register 0 data |
| OTP block 0 – factory segment | Partition base address + 81h to 84h | Factory-programmed data |
| OTP block 1 – user-programmable segment | Partition base address + 85h to 88h | User data |
| OTP lock register 1 | Partition base address + 89h | Lock register 1 data |
| OTP blocks 2–17 | Partition base address + 8Ah to 109h | User data |

Read CFI

Issuing the READ CFI command places the addressed partition in read CFI mode. Subsequent reads from that partition output CFI information.

The addressed partition remains in read CFI mode until a different READ command is issued, or a PROGRAM or ERASE operation is performed, or a BLOCK LOCK SETUP command is issued, which changes the read mode to read status register mode.

The READ CFI command functions independently of the voltage level on V_{pp} .

Issuing the READ CFI command to a partition that is actively programming or erasing changes that partition's read mode to read CFI mode. Subsequent reads from that partition will return invalid data until the PROGRAM or ERASE operation has completed.

After issuing a READ ID (0x90) or READ CFI (0x98) command to a partition, a READ ARRAY (0xFF) command must be issued to any partition address before reading the main array.

Note: After issuing a READ DEVICE INFORMATION (0x90) or CFI QUERY (0x98) command in any one of the partitions, a READ ARRAY (0xFF) command must be first be issued to any partition address before reading any portion of the main array.

Read Status Register

Issuing the READ STATUS REGISTER command places the addressed partition in read status register mode; other partitions are not affected. Subsequent reads from that partition output status register information.

Note: CE# or OE# must be toggled to update the status register data.

The addressed partition remains in read status register mode until a different READ MODE command is issued to that partition. Performing a PROGRAM, ERASE, or BLOCK LOCK operation also changes the partition's read mode to read status register mode.

The READ STATUS REGISTER command functions independently of the voltage level on V_{PP} .

Status register contents are valid only when $SR[7]=1$.

WAIT Operation

WAIT indicates the validity of output data during synchronous READ operations. It is asserted when output data is invalid and de-asserted when output data is valid. WAIT changes state only on valid clock edges. Upon power-up or exit from reset, WAIT defaults to LOW true ($RCR[10] = 0$).

WAIT is de-asserted during asynchronous reads. During WRITE operations, WAIT de-asserted on A/D-MUX devices.

Table 18: WAIT Behavior Summary – A/D MUX

| Device Operation | CE# | OE# | WE# | WAIT |
|-------------------------------|------|------|------|---|
| STANDBY (Device not selected) | HIGH | X | X | High-Z |
| OUTPUT DISABLE | LOW | HIGH | HIGH | De-asserted |
| Synchronous READ | LOW | LOW | HIGH | Active WAIT asserted = invalid data WAIT de-asserted = valid data |
| Asynchronous READ | LOW | LOW | HIGH | De-asserted |
| WRITE | LOW | HIGH | LOW | De-asserted |

Note: 1. This table does not apply to AADM devices. See AADM Mode for WAIT behavior in AADM mode.

Programming Modes

Each programming region in a block can be configured for either control mode or object mode.

The programming mode is automatically set based on the data pattern when a region is first programmed. Selecting either control mode or object mode is done according to the specific needs of the system. In control mode, code or data is frequently changed (such as the flash file system or header information). In object mode, large code or data (such as objects or payloads) is infrequently changed. By implementing the appropriate programming mode, software can efficiently organize how information is stored in the memory array.

Control mode programming regions and object mode programming regions can be intermingled within the same erase block. However, the programming mode of any region within a block can be changed only after erasing the entire block.

Control Mode

Control mode programming is invoked when only the A-half ($A3 = 0$) of the programming region is programmed to 0s. The B-half ($A3 = 1$) remains erased. Control mode allows up to 512 bytes of data to be programmed in the region. The information can be programmed in bits, bytes, or words.

Control mode supports the following programming methods:

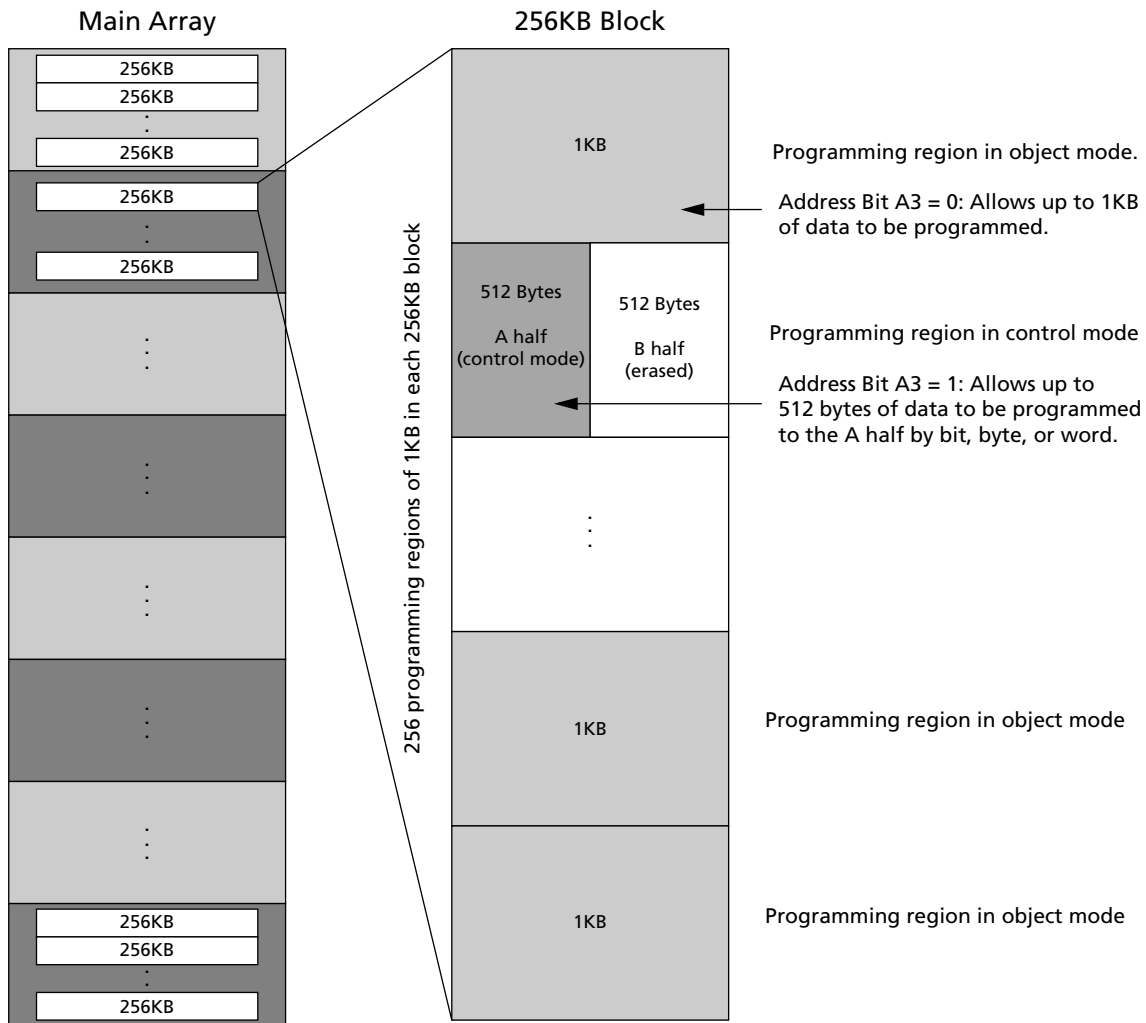
- Single-word programming (0041h)
- Buffered programming (00E9h/00D0h)
- Buffered enhanced factory programming (0080h/00D0h)

When buffered programming is used in control mode, all addresses must be in the A-half of the buffer ($A3 = 0$). During buffer fill, the B-half ($A3 = 1$) addresses do not need to be filled with 0xFFFF.

Control mode programming is useful for storing dynamic information, such as flash file system headers, file Info, and so on. Typically, control mode programming does not require the entire 512 bytes of data to be programmed at once. It may also contain data that is changed after initial programming using a technique known as “bit twiddling”. Header information can be augmented later with additional new information within a control-mode-programmed region. This allows implementation of legacy file systems, as well as transaction-based power-loss recovery.

In a control mode region, PROGRAM operations can be performed multiple times. However, care must be taken to avoid programming any zeros in the B-half ($A3 = 1$) of the region. Violation of this usage will cause SR[4] and SR[9] to be set, and the PROGRAM operation will be aborted.

Figure 7: Configurable Programming Regions: Control Mode and Object Mode



Object Mode

Object mode programming is invoked when one or more bits are programmed to zero in the B-half of the programming region (A3 = 1).

Object mode allows up to 1KB to be stored in a programming region. Multiple regions are used to store more than 1KB of information. If the object is less than 1KB, the unused content will remain as 0xFFFF (erased).

Object mode supports the following programming methods:

- Buffered programming (00E9h/00D0h)
- Buffered enhanced factory programming (0080h/00D0h)

Single-word programming (0041h) is not supported in object mode. To perform multiple PROGRAM operations within a programming region, control mode must be used. (Object mode is useful for storing static information, such as objects or payloads, that rarely change.)

Once the programming region is configured in object mode, it cannot be augmented or overwritten without first erasing the entire block containing the region. Subsequent PROGRAM operations to a programming region configured in object mode will cause SR[4] and SR[8] to be set and the PROGRAM operation to be aborted.

Issuing the 41h command to the B-half of an erased region will set error bits SR[8] and SR[9], and the PROGRAM operation will not proceed.

Figure 8: Configurable Programming Regions: Control Mode and Object Mode Segments

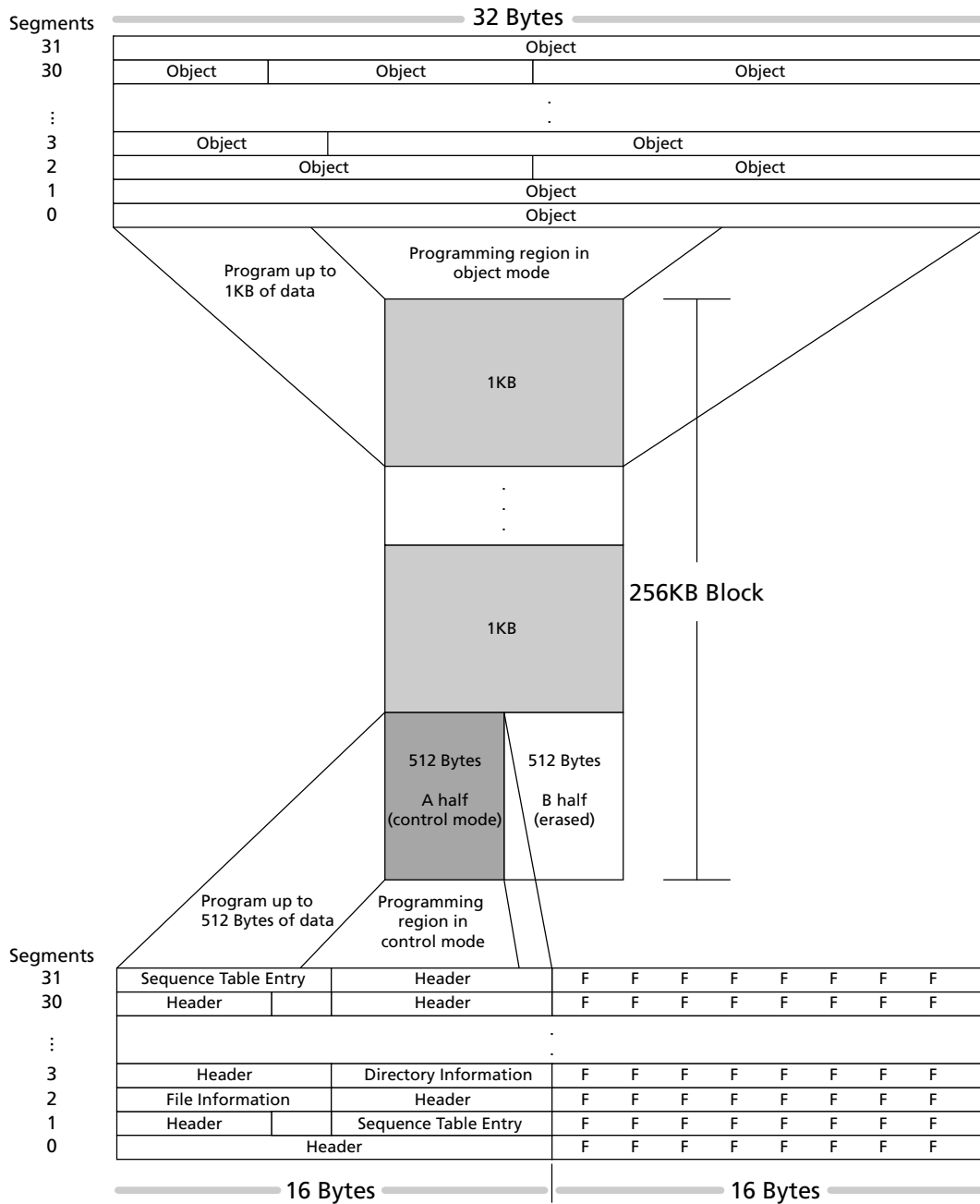


Table 19: Programming Region Next State

| Command Issued | Current State of Programming Region | | |
|--------------------------|--|--|---|
| | Erased | Control Mode | Object Mode |
| 0041h to B-half (A3 = 1) | Program fail; Illegal command SR[4,8,9] = 1 | Program fail; Illegal command SR[4,8,9] = 1 | Program fail; Illegal command SR[4,8,9] = 1 |
| 0041h to A-half (A3 = 0) | Program successful SR[4,8,9] = 0 Region configured to control mode | Program successful SR[4,8,9] = 0 | Program fail; Rewrite to object mode region SR[4,8] = 1 SR[9] = 0 |
| 00E9h to B-half (A3 = 1) | Program successful SR[4,8,9] = 0 Region configured to object mode | Program fail; Object data to control mode region SR[4,9] = 1 SR[8] = 0 | Program fail; Rewrite to object mode region SR[4,8] = 1 SR[9] = 0 |
| 00E9h to A-half (A3 = 0) | Program successful SR[4,8,9] = 0 Region configured to control mode | Program successful SR[4,8,9] = 0 | Program fail; Rewrite to object mode region SR[4,8] = 1 SR[9] = 0 |

Program Operations

Programming the array changes 1s to 0s. To change 0s to 1s, an ERASE operation must be performed. Only one PROGRAM operation can occur at a time. Programming is permitted during erase suspend.

Information is programmed into the array by issuing the appropriate command.

All PROGRAM operations require the addressed block to be unlocked and a valid V_{PP} voltage applied throughout the PROGRAM operation. Otherwise, the PROGRAM operation will abort, setting the appropriate status register error bit(s).

If the device is deselected during a PROGRAM or ERASE operation, the device continues to consume active power until the PROGRAM or ERASE operation has completed.

Table 20: PROGRAM Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-----------------------------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| SINGLE-WORD PROGRAM | Device address | 0041h | Device address | Array data |
| BUFFERED PROGRAM | Device address | 00E9h | Device address | 00D0h |
| BUFFERED ENHANCED FACTORY PROGRAM | Device address | 0080h | Device address | 00D0h |

Single-Word Programming

Single-word programming is performed by issuing the SINGLE-WORD PROGRAM command. This is followed by writing the desired data at the desired address. The read mode of the addressed partition is automatically changed to read status register mode, which remains in effect until another READ MODE command is issued.

Issuing the READ STATUS REGISTER command to another partition switches that partition's read mode to read status register mode, thereby enabling programming progress to be monitored from that partition's address.

Single-word programming is supported in control mode only. The array address specified must be in the A-half of the programming region.

During programming, the status register indicates a busy status ($SR[7] = 0$). Upon completion, the status register indicates a ready status ($SR[7] = 1$). The status register should be checked for any errors, then cleared.

The only valid command during programming is PROGRAM SUSPEND. After programming completes, any valid command can be issued.

Issuing the READ ARRAY, READ ID, or READ CFI command to a partition that is actively programming causes subsequent reads from that partition to output invalid data. Valid data is output only after the PROGRAM operation is complete.

Standby power levels are not realized until the PROGRAM operation has completed. Asserting RST# immediately aborts the PROGRAM operation, and array contents at the addressed location are indeterminate. The addressed block should be erased and the data reprogrammed.

Buffered Programming

Buffered programming programs multiple words simultaneously into the memory array. Data is first written to a programming buffer and then programmed into the array in buffer-sized increments, significantly reducing the effective word programming time.

Optimal performance and power consumption is realized only by aligning the starting address to buffer-sized boundaries within the array. Crossing a buffer-sized boundary can cause the buffered programming time to double.

The BUFFERED PROGRAM operation consists of the following fixed, predefined sequence of bus WRITE cycles: 1) Issue the SETUP command; 2) Issue a word count; 3) Fill the buffer with user data; and 4) Issue the CONFIRM command. Once the SETUP command has been issued to an address, subsequent bus WRITE cycles must use addresses within the same block throughout the operation; otherwise, the operation will abort. Bus READ cycles are allowed at any time and at any address.

Note: V_{PP} must be at V_{PPL} or V_{PPH} throughout the BUFFERED PROGRAM operation.

Upon programming completion, the status register indicates ready ($SR7 = 1$), and any valid command may be issued. A full status register check should be performed to check for any programming errors. If any error bits are set, the status register should be cleared using the CLEAR STATUS REGISTER command.

A subsequent BUFFERED PROGRAM operation can be initiated by issuing another SETUP command and repeating the buffered programming sequence. Any errors in the status register caused by a previous operation should first be cleared to prevent masking of errors that may occur during a subsequent BUFFERED PROGRAM operation.

Valid commands issued to the busy partition during array programming are READ STATUS and PROGRAM SUSPEND.

Issuing the READ ARRAY, READ ID, or READ CFI command to a partition that is actively programming causes subsequent reads from that partition to output invalid data. Valid data is output only after the PROGRAM operation has completed.

Buffered Enhanced Factory Programming

Buffered enhanced factory programming (BEFP) improves programming performance through the use of the write buffer, elevated programming voltage (V_{PPH}), and enhanced programming algorithm. User data is written into the write buffer, and then the buffer contents are automatically written into the array in buffer-sized increments.

Internal verification during programming (inherent to MLC technology) and status register error checking are used to determine proper completion of the PROGRAM operation. This eliminates delays incurred when switching between SINGLE-WORD PROGRAM and VERIFY operations.

BEFP consists of the following three distinct phases:

1. Setup phase: V_{PPH} and block lock checks
2. Program/verify phase: buffered programming and verification
3. Exit phase: block error check

BEFP is supported in both control mode and object mode. The programming mode selection for the entire array block is driven by the specific type of information, such as header or object data. Header/object data is aligned on a 1KB programming region boundary in the main array block.

Table 21: BEFP Requirements and Considerations

| | |
|----------------------------|---|
| BEFP Requirements | Temperature (T_{AMBIENT}) must be 25 °C, \pm 5 °C |
| | Voltage on V_{CC} must be within the allowable operating range |
| | Voltage on V_{PP} must be within the allowable operating range |
| | Block being programmed must be erased and unlocked |
| BEFP Considerations | Block cycling below 100 ERASE cycles |
| | BEFP programs within one block at a time |
| | BEFP cannot be suspended |

BEFP Setup Phase

Issuing the BEFP SETUP and CONFIRM command sequence starts the BEFP algorithm. The read mode of the addressed partition is automatically changed to read status register mode.

The address used when issuing the SETUP and CONFIRM commands must be buffer-size aligned within the block being programmed; buffer contents cannot cross block boundaries.

Note: The READ STATUS REGISTER command must not be issued; it will be interpreted as data to be written to the write buffer.

A setup delay ($t_{\text{BEFP/setup}}$) occurs while the internal algorithm checks V_{PP} and block lock status. If errors are detected, the appropriate status register error bits are set and the operation aborts.

The status register should be polled for successful BEFP setup, indicated by $\text{SR}[7:0] = 0$ (device busy, buffer ready for data).

BEFP Program/Verify Phase

Data is first written into the write buffer, then programmed into the array. During the buffer fill sequence, the address used must be buffer-size aligned. Use of any other address will cause the operation to abort with a program fail error, and any data previously loaded in the buffer will not be programmed into the array.

The buffer fill data is stored in sequential buffer locations starting at address 00h. A word count equal to the maximum buffer size is used; therefore, the buffer must be completely filled. If the amount of data is less than the maximum buffer size, the remaining buffer locations must be padded with FFFFh to completely fill the buffer.

Array programming starts as soon as the write buffer is full. Data words from the write buffer are programmed into sequential array locations. $\text{SR}_0 = 1$ indicates the write buffer is not available while the BEFP algorithm programs the array.

The status register should be polled for $\text{SR}_0 = 0$ (buffer ready for data) to determine when the array programming has completed and the write buffer is again available for loading. The internal address is automatically incremented to enable subsequent array programming to continue from where the previous buffer-fill/array program sequence ended within the block. This cycle can be repeated to program the entire block.

BEFP Exit Phase

To exit the program/verify phase, write FFFFh to an address outside of the block.

The status register should be polled for SR7 = 1 (device ready), indicating the BEFP algorithm has finished running and the device has returned to normal operation.

A full status register error check should be performed to ensure the block was programmed successfully.

Erase Operations

BLOCK ERASE

Erasing a block changes 0s to 1s. To change 1s to 0s, a PROGRAM operation must be performed. Erasing is performed on a block basis; an entire block is erased each time an erase command sequence is issued. Once a block is fully erased, all addressable locations within that block read as logical 1s (FFFFh).

Only one BLOCK ERASE operation can occur at a time. A BLOCK ERASE operation is not permitted during program suspend. All BLOCK ERASE operations require the addressed block to be unlocked, and V_{PP} must be at V_{PPL} or V_{PPH} throughout the BLOCK ERASE operation. Otherwise, the operation aborts, setting the appropriate status register error bit(s).

To perform a BLOCK ERASE operation, issue the BLOCK ERASE SETUP command at the desired block address. The read mode of the addressed partition automatically changes to read status register mode and remains in effect until another READ MODE command is issued.

The ERASE CONFIRM command latches the address of the block to be erased. The addressed block is preconditioned (programmed to all 0s), erased, and then verified.

Issuing the READ STATUS REGISTER command to another partition switches that partition's read mode to the read status register, thereby allowing block erase progress to be monitored from that partition's address. SR0 indicates whether the addressed partition or the other partition is erasing.

During a BLOCK ERASE operation, the status register indicates a busy status ($SR[7] = 0$).

Issuing the READ ARRAY command to a partition that is actively erasing a main block causes subsequent reads from that partition to output invalid data. Valid array data is output only after the BLOCK ERASE operation has finished.

Upon completion, the status register indicates a ready status ($SR[7] = 1$). The status register should be checked for any errors, and then cleared.

If the device is deselected during an ERASE operation, the device continues to consume active power until the ERASE operation is completed.

Asserting RST# immediately aborts the BLOCK ERASE operation, and array contents at the addressed location are indeterminate. The addressed block should be erased again.

The only valid command during a BLOCK ERASE operation is ERASE SUSPEND. After the BLOCK ERASE operation has completed, any valid command can be issued.

Table 22: ERASE Command Bus Cycle

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| BLOCK ERASE | Device address | 0020h | Block address | 00D0h |

SUSPEND and RESUME Operations

PROGRAM and ERASE operations of the main array can be suspended to perform other device operations, and then subsequently resumed. OTP area programming operations cannot be suspended. During erase suspend or program suspend, the addressed block must remain unlocked, V_{PP} must be at V_{PPL} or V_{PPH} , and $WP\#$ must remain unchanged. Otherwise, the ERASE or PROGRAM operation will abort, setting the appropriate status register error bit(s).

SUSPEND Operation

To suspend an ongoing ERASE or PROGRAM operation, issue the SUSPEND command to any device address. Issuing the SUSPEND command does not change the read mode.

Upon issuing a SUSPEND command, the ongoing ERASE or PROGRAM operation suspends after a delay of t_{SUSP} . The operation is suspended only when $SR[7:6] = 1$ (erase suspend) or $SR[7:2] = 1$ (program suspend).

While suspended, reading from a block that was being erased or programmed is not allowed. Also, programming within an erase suspended block is not allowed, and if attempted, will result in a programming error ($SR[4] = 1$). Erasing under program suspend is not allowed. However, array programming under erase suspend is allowed, and can also be suspended. This results in a simultaneous erase suspend and program suspend condition, indicated by $SR[7:6,2] = 1$. Additional valid commands while suspended are READ ARRAY, READ STATUS REGISTER, READ ID, READ CFI, CLEAR STATUS REGISTER, and RESUME. No other commands are allowed.

During suspend, $CE\#$ may be de-asserted, placing the device in standby and reducing active current to standby levels. V_{PP} must remain at V_{PPL} or V_{PPH} , and $WP\#$ must remain unchanged.

Asserting $RST\#$ aborts any suspended BLOCK ERASE and PROGRAM operations; array contents at the addressed locations will be indeterminate.

During suspend, $CE\#$ may be de-asserted. The device is placed in standby, reducing active current. V_{PP} must remain at V_{PPL} or V_{PPH} , and $WP\#$ must remain unchanged.

Asserting $RST\#$ aborts suspended BLOCK ERASE and PROGRAM operations; array contents at the addressed locations are indeterminate.

A BUFFERED PROGRAM command sequence can be terminated during a the BUFFER FILL operation while in an erase suspend by issuing any non-00D0h command (non-confirm command) to the same block address to which the BUFFERED PROGRAM command was written.

Table 23: Valid Commands During Suspend

| Device Command | Program Suspend | Erase Suspend |
|-------------------------|-----------------|---------------|
| READ ARRAY | Allowed | Allowed |
| READ STATUS REGISTER | Allowed | Allowed |
| CLEAR STATUS REGISTER | Allowed | Allowed |
| READ DEVICE INFORMATION | Allowed | Allowed |
| CFI QUERY | Allowed | Allowed |
| WORD PROGRAM | Not Allowed | Allowed |

Table 23: Valid Commands During Suspend (Continued)

| Device Command | Program Suspend | Erase Suspend |
|-----------------------------------|-----------------|---------------|
| BUFFERED PROGRAM | Not Allowed | Allowed |
| BUFFERED ENHANCED FACTORY PROGRAM | Not Allowed | Not Allowed |
| BLOCK ERASE | Not Allowed | Not Allowed |
| PROGRAM/ERASE SUSPEND | Not Allowed | Not Allowed |
| PROGRAM/ERASE RESUME | Allowed | Allowed |

RESUME Operation

To resume a suspended ERASE or PROGRAM operation, issue the RESUME command to any device address. The ERASE or PROGRAM operation continues where it left off, and the respective status register suspend bit is cleared. Issuing the RESUME command does not change the read mode.

When the RESUME command is issued during a simultaneous erase suspend or program suspend condition, the PROGRAM operation is resumed first. Upon completion of the PROGRAM operation, the status register should be checked for any errors, and cleared if needed. The RESUME command must be issued again to complete the ERASE operation. Upon completion of the ERASE operation, the status register should be checked for any errors, and cleared if needed.

Table 24: SUSPEND and RESUME Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|---------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| SUSPEND | Device address | 00B0h | – | – |
| RESUME | Device address | 00D0h | – | – |

BLANK CHECK Operation

Blank check verifies whether a main-array block is completely erased. A BLANK CHECK operation is performed one block at a time, and cannot be used during program suspend or erase suspend.

To use blank check, first issue the BLANK CHECK SETUP command followed by the CONFIRM command. The read mode of the addressed partition is automatically changed to read status register mode, which remains in effect until another read mode is issued.

During a BLANK CHECK operation, the status register indicates a busy status (SR[7] = 0). Upon completion, the status register indicates a ready status (SR[7] = 1). Issuing the READ STATUS REGISTER command to another partition switches that partition's read mode to read status register mode, thereby allowing the BLANK CHECK operation to be monitored from that partition's address.

The status register should be checked for any errors, and then cleared. If the BLANK CHECK operation fails (the block is not completely erased), then the status register will indicate a blank check error (SR[7:5] = 1).

The only valid command during a BLANK CHECK operation is read status. Blank check cannot be suspended. After the BLANK CHECK operation has completed, any valid command can be issued.

Table 25: BLANK CHECK Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| BLANK CHECK | Block address | 00BCh | Block address | 00D0h |

Block Lock

Two methods of block lock control are available: software and hardware. Software control uses the BLOCK LOCK and BLOCK UNLOCK commands. Hardware control uses the BLOCK LOCK-DOWN command along with asserting WP#.

Upon power-up or exit from reset, all main array blocks are locked, but not locked down. Locked blocks cannot be erased or programmed. BLOCK LOCK and UNLOCK operations are independent of the voltage level on V_{pp}.

To lock, unlock, or lock-down a block, first issue the SETUP command to any address within the desired block. The read mode of the addressed partition is automatically changed to read status register mode. Next, issue the desired CONFIRM command to the block's address. Note that the CONFIRM command determines the operation performed. The status register should be checked for any errors, and then cleared.

The lock status of a block can be determined by issuing the READ ID command, and then reading from the block's base address + 02h. See the table below table for the lock-bit settings.

Blocks cannot be locked or unlocked while being actively programmed or erased. Blocks can be locked or unlocked during erase suspend, but not during program suspend. If a BLOCK ERASE operation is suspended, and then the block is locked or locked down, the lock status of the block will be changed immediately. When resumed, the ERASE operation will still complete.

Block lock-down protection is dependent on WP#. A locked-down block can only be unlocked by issuing the BLOCK UNLOCK command with WP# de-asserted. To return an unlocked block to the locked-down state, a BLOCK LOCK-DOWN command must be issued prior to asserting WP#.

When WP# = V_{IL}, blocks locked down are locked, and cannot be unlocked using the BLOCK UNLOCK command.

When WP# = V_{IH}, block lock-down protection is disabled; locked-down blocks can be individually unlocked using the BLOCK UNLOCK command.

Subsequently, when WP# = V_{IL}, previously locked-down blocks are once again locked and locked-down, including locked-down blocks that may have been unlocked while WP# was de-asserted.

Issuing the BLOCK LOCK-DOWN command to an unlocked block does not lock the block. However, asserting WP# after issuing the BLOCK LOCK-DOWN command locks (and locks down) the block. Lock-down for all blocks is only cleared upon power-up or exit from reset.

Table 26: BLOCK LOCK Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|--------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| BLOCK LOCK | Block address | 0060h | Block address | 0001h |
| BLOCK UNLOCK | Block address | 0060h | Block address | 00D0h |

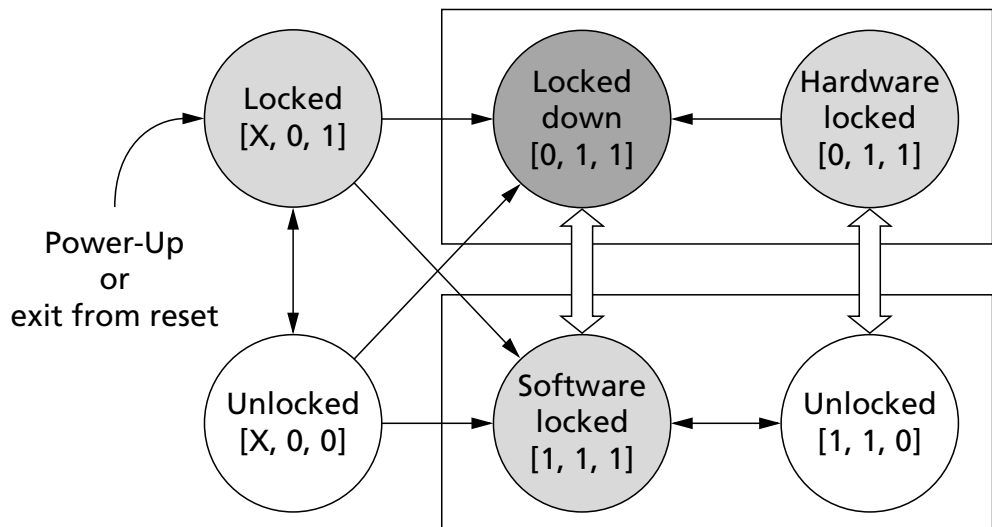
Table 26: BLOCK LOCK Command Bus Cycles (Continued)

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|-----------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| BLOCK LOCK-DOWN | Block address | 0060h | Block address | 002Fh |

Table 27: Block Lock Configuration

| Block Lock Configuration | Block Base Address | Bit |
|--------------------------|---------------------------|-----------|
| Block is unlocked | Block base address = 0x02 | DQ0 = 0b0 |
| Block is locked | Block base address = 0x02 | DQ0 = 0b1 |
| Block is not locked down | Block base address = 0x02 | DQ1 = 0b0 |
| Block is locked down | Block base address = 0x02 | DQ1 = 0b1 |

Figure 9: BLOCK LOCK Operations



↔ Software control (LOCK, UNLOCK, LOCK-DOWN command)

⇔ Hardware control (WP#)

- Notes:
1. The [n,n,n] denotes logical state of WP#, DQ1, and DQ0, respectively; X = "Don't Care."
 2. The [0,1,1] states should be tracked by system software to differentiate between the hardware-locked state and the lock-down state.

One-Time Programmable Operations

The device contains sixteen 128-bit one-time programmable (OTP) blocks, two 64-bit OTP blocks, and two 16-bit OTP lock registers. OTP lock register 0 is used for locking OTP blocks 0 and 1 (two 64-bit blocks), and OTP lock register 1 is used for locking OTP blocks 2 through 17 (sixteen 128-bit blocks).

Each block contains OTP bits that are factory set to 1 and can only be programmed from 1 to 0; OTP block bits cannot be erased from 0 back to 1. This feature makes the OTP blocks particularly useful for implementing system-level security schemes, permanently storing data, or storing fixed system parameters.

OTP block 0 is pre-programmed with a unique 64-bit value and locked at the factory. OTP block 1 contains all 1s and is user-programmable. OTP blocks 1 through 16 contain all 1s and are user-programmable.

Each OTP block can be accessed multiple times to program individual bits, as long as the block remains unlocked. When a lock register bit is programmed, the associated OTP block can only be read—it can no longer be programmed.

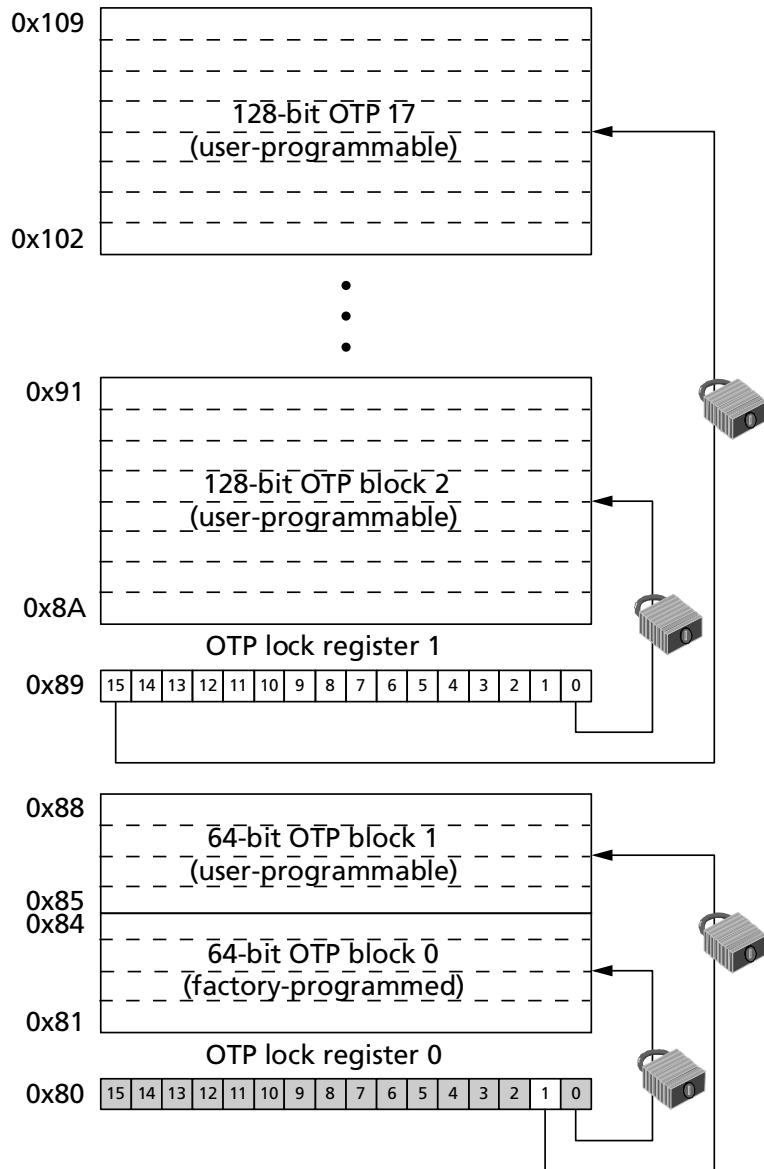
OTP lock register bits lock out subsequent programming of the corresponding OTP block. Each OTP block can be locked by programming its corresponding lock bit to 0. As long as an OTP block remains unlocked (that is, its lock bit = 1), any of its remaining 1 bits can be programmed to 0.

Note: Once an OTP block is locked, it cannot be unlocked. Attempts to program a locked OTP block will fail with error bits set. Additionally, because the lock register bits themselves are OTP, when programmed, lock register bits cannot be erased. Therefore, when an OTP block is locked, it cannot be unlocked.

Table 28: Program OTP Area Command Bus Cycles

| Command | Setup WRITE Cycle Address Bus | Setup WRITE Cycle Data Bus | Confirm WRITE Cycle Address Bus | Confirm WRITE Cycle Data Bus |
|------------------|-------------------------------|----------------------------|---------------------------------|------------------------------|
| PROGRAM OTP AREA | Device address | 00C0h | OTP register address | Register data |

Figure 10: OTP Area Map



Programming OTP Area

OTP area programming is performed 16 bits at a time; only zeros within the data word affect any change to the OTP bits.

To program any OTP blocks or lock registers, first issue the PROGRAM OTP AREA SETUP command at any device address. The read state of that partition changes to read status. Next, write the desired OTP data at the desired OTP address.

Attempting to program outside of the OTP area causes a program error (SR[4] = 1).

Attempting to program a locked OTP block causes a program error and a lock error (SR[4] = 1, SR[1] = 1).

OTP area programming cannot be suspended.

Reading OTP Area

The OTP area is read from within the address space of any partition. To read from the OTP area, the following must be done:

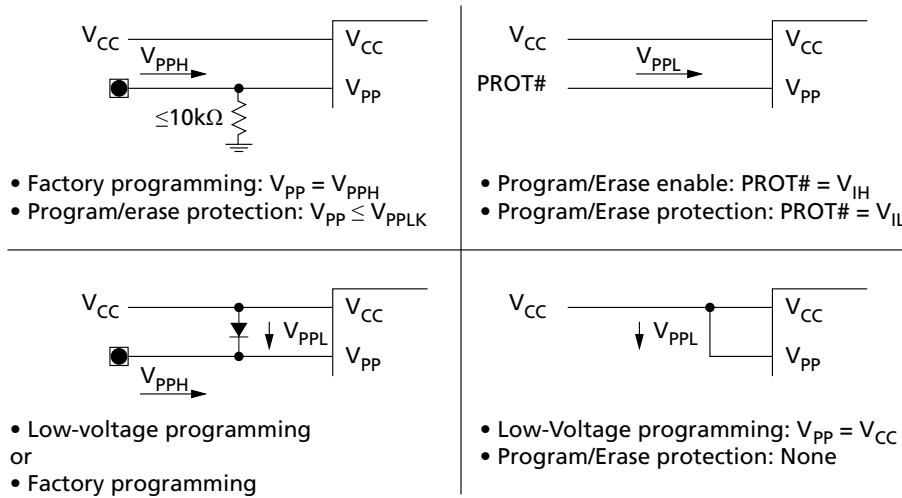
1. Issue the READ ID command at the address of any partition to place that partition in the read ID state.
2. Perform a READ operation at the base address of that partition, plus the address offset corresponding to the OTP word to be read. Data is read 16 bits at a time.

Global Main-Array Protection

Global main-array protection can be implemented by controlling V_{pp} . When programming or erasing main-array blocks, V_{pp} must be equal to or greater than $V_{PPL, min}$. When V_{pp} is below V_{PPLK} , PROGRAM or ERASE operations are inhibited, thus providing absolute protection of the main array.

Various methods exist for controlling V_{pp} , ranging from simple logic control to off-board voltage control. The following figure shows example V_{pp} supply connections that can be used to support PROGRAM or ERASE operations and main-array protection.

Figure 11: V_{pp} Supply Connection Example



Power and Reset Specifications

Initialization

Proper device initialization and operation is dependent on the power-up/down sequence, reset procedure, and adequate power-supply decoupling.

Power-Up and Down

To avoid conditions that may result in spurious PROGRAM or ERASE operations, the power sequences shown below are recommended. Note that each power supply must be at its minimum voltage range before applying or removing the next supply voltage in the sequence. Also, device inputs must not be driven until all supply voltages have attained their minimum range, and RST# should be LOW during all power transitions.

When powering down the device, voltages should reach 0V before power is reapplied to ensure proper device initialization. Otherwise, indeterminate operation could result.

When V_{CCQ} goes below V_{LKOQ} , the device is reset.

Table 29: Power Sequencing

| Power Supply | Power-Up Sequence | | | | Power-Down Sequence | | | |
|---------------|-------------------|---------------------|--------------------|--------------------|---------------------|--------------------|---------------------|---------------------|
| $V_{CC,min}$ | First | First | First ¹ | First ¹ | Third | Second | Second ¹ | Second ¹ |
| $V_{CCQ,min}$ | Second | Second ¹ | First ¹ | Second | Second | First ¹ | Second ¹ | First |
| $V_{PP,min}$ | Third | Second ¹ | Second | First ¹ | First | First ¹ | First | Second ¹ |

Note: 1. Connected/sequenced together.

Reset

During power-up and power-down, RST# should be asserted to prevent spurious PROGRAM or ERASE operations. While RST# is LOW, device operations are disabled, all inputs such as address and control are ignored, and all outputs such as data and WAIT are placed in High-Z. Invalid bus conditions are effectively masked out.

Upon power-up, RST# can be de-asserted after t_{VCCPH} , allowing the device to exit from reset. Upon exiting from reset, the device defaults to asynchronous read array mode, and the status register defaults to 0080h. Array data is available after t_{PHQV} , or a bus WRITE cycle can begin after t_{PHWL} . If RST# is asserted during a PROGRAM or ERASE operation, the operation will abort and array contents at that location will be invalid.

For proper system initialization, connect RST# to the LOW true reset signal that asserts whenever the processor is reset. This will ensure the device is in the expected read mode (read array) upon startup.

Figure 12: RESET Operation Waveforms

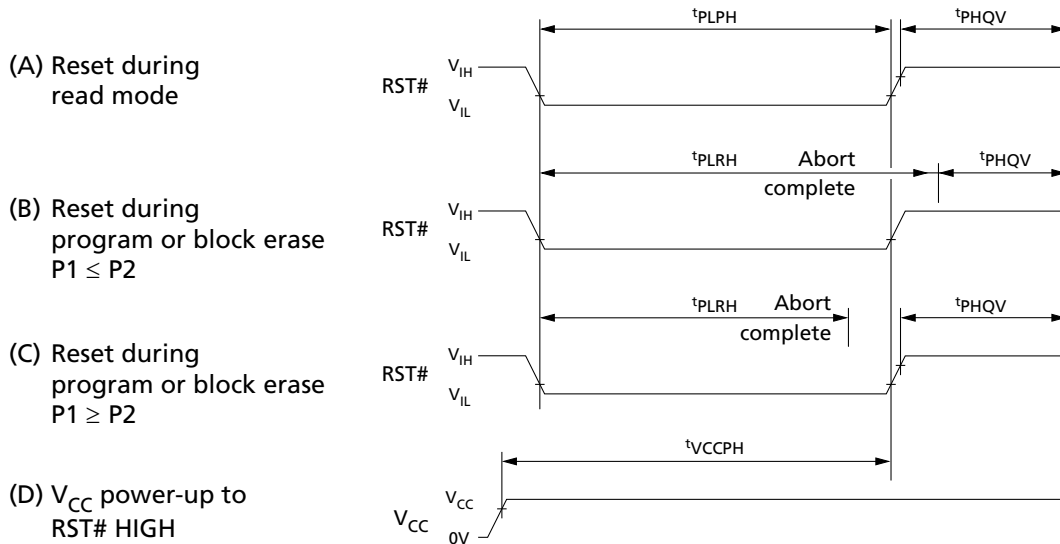


Table 30: Reset Specifications

Note 1 applies to all

| Parameter | Symbol | Min | Max | Unit | Notes |
|--|-------------|-----|-----|---------|---------|
| RST# pulse width LOW | t_{PLPH} | 100 | – | ns | 2, 3, 6 |
| RST# LOW to device reset during erase | t_{PLRH} | – | 25 | μs | 3, 6 |
| RST# LOW to device reset during program | | – | 25 | | 3, 6 |
| V_{CC} power valid to RST# de-assertion (HIGH) | t_{VCCPH} | 300 | – | | 4, 5 |

- Notes:
1. These specifications are valid for all packages and speeds, and are sampled, not 100% tested.
 2. The device might reset if t_{PLPH} is $< t_{PLPH}$ MIN, but this is not guaranteed.
 3. Not applicable if RST# is tied to V_{CCQ} .
 4. If RST# is tied to the V_{CC} supply, the device is not ready until t_{VCCPH} after $V_{CC} \geq V_{CC,min}$.
 5. If RST# is tied to any supply/signal with V_{CCQ} voltage levels, the RST# input voltage must not exceed V_{CC} until $V_{CC} \geq V_{CC,min}$.
 6. Reset completes within t_{PLPH} if RST# is asserted while no ERASE or PROGRAM operation is executing.

Automatic Power Saving

Automatic power saving provides low-power operation following reads during active mode. After data is read from the memory array and the address lines are quiescent, automatic power savings automatically places the device into standby. In automatic power savings, device current is reduced to I_{CCAPS} .

Power Supply Decoupling

Flash memory devices require careful power supply decoupling to prevent external transient noise from affecting device operations, and to prevent internally generated transient noise from affecting other devices in the system.

Ceramic chip capacitors of 0.01 μ F to 0.1 μ F should be used between all V_{CC} , V_{CCQ} , and V_{PP} supply connections and system ground. These high-frequency, inherently low-inductance capacitors should be placed as close as possible to the device package, or on the opposite side of the printed circuit board close to the center of the device package footprint.

Larger (4.7 μ F to 33.0 μ F) electrolytic or tantalum bulk capacitors should also be distributed as needed throughout the system to compensate for voltage sags and surges caused by circuit trace inductance.

Transient current magnitudes depend on the capacitive and inductive loading on the device's outputs. For best signal integrity and device performance, high-speed design rules should be used when designing the printed-circuit board. Circuit-trace impedances should match output-driver impedance with adequate ground-return paths. This will help minimize signal reflections (overshoot/undershoot) and noise caused by high-speed signal edge rates.

Electrical Specifications

Stresses greater than those listed may cause permanent damage to the device. This is a stress rating only. Exposure to absolute maximum rating and operating conditions for extended periods may adversely affect reliability. Stressing the device beyond the absolute maximum ratings may cause permanent damage. These are stress ratings only.

Table 31: Absolute Maximum Ratings

| Parameter | Min | Max | Units | Notes |
|--|---------|-----------------|--------|---------|
| Temperature under bias (T_A) | -40 | 105 | °C | 5 |
| Storage temperature (T_A) | -65 | 125 | °C | 5 |
| V_{PP} voltage | -2.0 | 11.5 | V | 1, 2, 3 |
| V_{CC} voltage | -2.0 | $V_{CCQ} + 2.0$ | V | 1 |
| Voltage on any input/output signal (except V_{CC} , V_{CCQ} , and V_{PP}) | -2.0 | $V_{CCQ} + 2.0$ | V | 2 |
| V_{CCQ} voltage | -0.2 | $V_{CCQ} + 2.0$ | V | 1 |
| V_{PPH} time | - | 80 | hours | 3 |
| Output short circuit current | - | 100 | mA | 4 |
| Block PROGRAM/ERASE cycles: Main blocks | 100,000 | - | Cycles | 3 |

- Notes:
1. Voltages shown are specified with respect to V_{SS} . During transitions, the voltage potential between V_{SS} and input/output and supply pins may undershoot to -1.0V for periods less than 20ns and may overshoot to $V_{CCQ}(MAX) + 1.0V$ for periods less than 20ns.
 2. Voltages shown are specified with respect to V_{SS} . During transitions, the voltage potential between V_{SS} and supply pins may undershoot to -2.0V for periods less than 20ns and may overshoot to $V_{CC}(MAX) + 2.0V$ for periods less than 20ns.
 3. Operation beyond this limit may degrade performance.
 4. Output shorted for no more than one second; no more than one output shorted at a time.
 5. Temperature specified is ambient (T_A), not case (T_C).

Table 32: Operating Conditions

| Symbol | Parameter | Min | Max | Units | Notes |
|-----------|---------------------------------------|-----|-----|-------|-------|
| T_A | Operating temperature | -40 | 105 | °C | 1 |
| V_{CC} | V_{CC} supply voltage | 1.7 | 2.0 | V | |
| V_{CCQ} | I/O supply voltage | 1.7 | 2.0 | V | |
| V_{PPL} | V_{PP} voltage supply (logic level) | 0.9 | 2.0 | V | |
| V_{PPH} | Factory programming V_{PP} | 8.5 | 9.5 | V | |

- Note: 1. Temperature specified is ambient (T_A), not case (T_C).



Electrical Specifications – DC Current and Voltage Characteristics and Operating Conditions

Table 33: DC Current Characteristics and Operating Conditions ($V_{CCQ} = 1.7V-2.0V$)

| Parameter | Symbol | Conditions | Density (Mb) | Typ | Max | Unit | Notes |
|--|--------------------------|--|--------------------|----------------|-------------------|---------|---------|
| Input Load, Output Leakage, Standby | | | | | | | |
| Input load current | I_{LI} | $V_{CC} = V_{CC,max}; V_{CCQ} = V_{CCQ,max}; V_{IN} = V_{CCQ}$ or V_{SS} | – | – | ± 1 | μA | 1 |
| Output leakage current | I_{LO} | $V_{CC} = V_{CC,max}; V_{CCQ} = V_{CCQ,max}; V_{IN} = V_{CCQ}$ or V_{SS} | – | – | ± 1 | μA | 1 |
| V_{CC} standby | I_{CCS} , I_{CCD} | $V_{CC} = V_{CC,max}; V_{CCQ} = V_{CCQ,max}; CE\# = V_{CCQ}; RST\# = V_{CCQ}$ or GND (for I_{CCS}); WP# = V_{IH} | 256 512 1024 | 50 60 70 | 280 280 280 | μA | 1, 2 |
| Average V_{CC} Read | | | | | | | |
| Average V_{CC} read current; Asynchronous single-word read; f = 5 MHz; 1 CLK | I_{CCR} | $V_{CC} = V_{CC,max}; CE\# = V_{IL}; OE\# = V_{IH};$ Inputs: V_{IL} or V_{IH} | – | 25 | 30 | mA | 1, 3 |
| Average V_{CC} read current; f = 13 MHz; 17 CLK; Burst = 16-word | I_{CCR} | $V_{CC} = V_{CC,max}; CE\# = V_{IL}; OE\# = V_{IH};$ Inputs: V_{IL} or V_{IH} | – | 11 | 15 | mA | 1, 3 |
| Average V_{CC} read current; Synchronous burst read; f = 66 MHz; LC = 7; Burst = 8-word; Burst = 16-word; Burst = Continuous | I_{CCR} | $V_{CC} = V_{CC,max}; CE\# = V_{IL}; OE\# = V_{IH};$ Inputs: V_{IL} or V_{IH} | – | 22 19 25 | 32 26 34 | mA | 1, 3 |
| Average V_{CC} read current; Synchronous burst read; f = 108 MHz; LC = 10; Burst = 8-word; Burst = 16-word; Burst = Continuous | I_{CCR} | $V_{CC} = V_{CC,max}; CE\# = V_{IL}; OE\# = V_{IH};$ Inputs: V_{IL} or V_{IH} | – | 26 23 32 | 36 30 42 | mA | 1, 3 |
| Average V_{CC} read current; Synchronous burst read; f = 133 MHz; LC = 13; Burst = 8-word; Burst = 16-word; Burst = Continuous | I_{CCR} | $V_{CC} = V_{CC,max}; CE\# = V_{IL}; OE\# = V_{IH};$ Inputs: V_{IL} or V_{IH} | – | 29 24 40 | 35 33 46 | mA | 1, 3 |
| V_{CC} Program, Erase, Blank Check | | | | | | | |
| V_{CC} Program | I_{CCW} | $V_{PP} = V_{PPL}$ or $V_{PP} = V_{PPH}$ | – | 45 | 60 | mA | 1, 3, 4 |
| V_{CC} Erase | I_{CCE} | Program/erase in progress | – | 30 | 66 | mA | 1, 3, 4 |

Table 33: DC Current Characteristics and Operating Conditions ($V_{CCQ} = 1.7V-2.0V$) (Continued)

| Parameter | Symbol | Conditions | Density (Mb) | Typ | Max | Unit | Notes |
|--|---|--|--------------------|----------------|-------------------|---------|---------|
| V_{CC} Blank Check | I_{CCBC} | $V_{PP} = V_{PPL}$ or $V_{PP} = V_{PPH}$; Blank check in progress | – | 40 | 65 | mA | 1, 3 |
| V_{CC} Program suspend V_{CC} Erase suspend | I_{CCWS} , I_{CCES} | $CE\# = V_{CCQ}$; Suspend in progress | 256 512 1024 | 50 60 70 | 280 280 280 | μA | 1, 3, 5 |
| V_{PP} Program, Read, Erase, Blank Check, Standby | | | | | | | |
| V_{PP} standby current; V_{PP} program suspend current; V_{PP} erase suspend current | I_{PPS} , I_{PPWS} , I_{PPES} | $V_{PP} = V_{PPL}$; Suspend in progress | – | 0.2 | 5 | μA | 3 |
| V_{PP} read | I_{PPR} | $V_{PP} \leq V_{CC}$ | – | 2 | 15 | μA | 3 |
| V_{PP} program current | I_{PPW} | $V_{PP} = V_{PPL} = V_{PPH}$; Program in progress | – | 0.05 | 0.10 | mA | 3 |
| V_{PP} erase current | I_{PPE} | $V_{PP} = V_{PPL} = V_{PPH}$; Erase in progress | – | 0.05 | 0.10 | mA | 3 |
| V_{PP} blank check current | I_{PPBC} | $V_{PP} = V_{PPL} = V_{PPH}$; Blank check in progress | – | 0.05 | 0.10 | mA | 3 |
| Automatic Power Savings | | | | | | | |
| Automatic power savings | I_{CCAPS} | $V_{CC} = V_{CC,max}$; $V_{CCQ} = V_{CCQ,max}$; $CE\# = V_{SSQ}$; $RST\# = V_{CCQ}$; All inputs are at rail-to-rail (V_{CCQ} or V_{SSQ}) | 256 512 1024 | 50 60 70 | 280 280 280 | μA | – |

- Notes:
1. All currents are RMS unless noted. Typical values at typical V_{CCQ} , $T_C = +25^\circ C$.
 2. I_{CCS} is the average current measured over any 5ms time interval $5\mu s$ after $CE\#$ is de-asserted.
 3. Sampled, not 100% tested.
 4. I_{CCW} , I_{CCE} is measured over typical or max times specified in Program and Erase Characteristics.
 5. I_{CCES} is specified with the device deselected. If the device is read while in erase suspend, current is $I_{CCES} + I_{CCR}$.



Table 34: DC Voltage Characteristics and Operating Conditions ($V_{CCQ} = 1.7V-2.0V$)

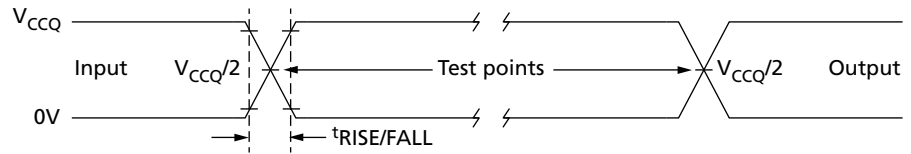
| Parameter | Symbol | Conditions | Min | Max | Unit | Notes |
|--------------------------|------------|---|------------------|-----------|------|-------|
| Input low voltage | V_{IL} | – | 0 | 0.45 | V | 1 |
| Input high voltage | V_{IH} | – | $V_{CCQ} - 0.45$ | V_{CCQ} | V | 1 |
| Output low voltage | V_{OL} | $V_{CC} = V_{CC,min}; V_{CCQ} = V_{CCQ,min}; I_{OL} = 100\mu A$ | – | 0.1 | V | |
| Output high voltage | V_{OH} | $V_{CC} = V_{CC,min}; V_{CCQ} = V_{CCQ,min}; I_{OL} = 100\mu A$ | $V_{CCQ} - 0.1$ | – | V | |
| V_{PP} lockout voltage | V_{PPLK} | – | – | 0.4 | V | 2 |
| V_{CC} lock voltage | V_{LKO} | – | 1.0 | – | V | |
| V_{CCQ} lock voltage | V_{LKOQ} | – | 0.8 | – | V | |

- Notes:
1. Input voltages can undershoot to $-1.0V$ and overshoot to $V_{CCQ} + 1V$ for durations of 2ns or less.
 2. $V_{PP} < V_{PPLK}$ inhibits ERASE and PROGRAM operations. Do not use V_{PPL} and V_{PPH} outside of their valid ranges.

Electrical Specifications – AC Characteristics and Operating Conditions

AC Test Conditions

Figure 13: AC Input/Output Reference Waveform



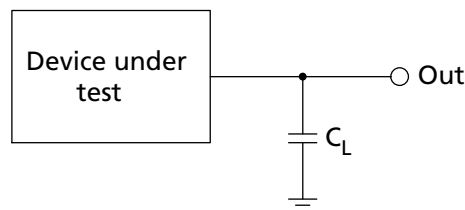
Note: 1. AC test inputs are driven at V_{CCQ} for Logic 1, and 0.0V for Logic 0. Input/output timing begins/ends at $V_{CCQ}/2$. Input rise and fall times (10% to 90%) <5ns. Worst-case speed occurs at $V_{CC} = V_{CC,min}$.

Table 35: AC Input Requirements

| Parameter | Symbol | Frequency | Min | Max | Unit | Condition | Notes |
|--|-----------------|-------------------|-----|-----|------|--|-------|
| Inputs rise/fall time (Address, CLK, CE#, OE#, ADV#, WE#, WP#) | $t_{RISE/FALL}$ | @133 MHz, 108 MHz | 0.3 | 1.2 | ns | V_{IL} to V_{IH} or V_{IH} to V_{IL} | – |
| | | @66 MHz | 0 | 3 | | | |
| Address-address skew | t_{ASKW} | | 0 | 3 | ns | @ $V_{CCQ}/2$ | 1 |

Note: 1. For an address to be latched the skew is defined as the time when the first address bit is valid to the last address bit going valid.

Figure 14: Transient Equivalent Testing Load Circuit



Notes: 1. See Test Configuration Load Capacitor Values for Worst Case Speed Conditions table for component values for the test configurations.
 2. C_L includes jig capacitance.

Table 36: Test Configuration Load Capacitor Values for Worst Case Speed Conditions

| Test Configuration | C_L (pF) |
|--------------------|------------|
| 1.7V Standard test | 30 |
| 2.0V Standard test | 30 |

Figure 15: Clock Input AC Waveform

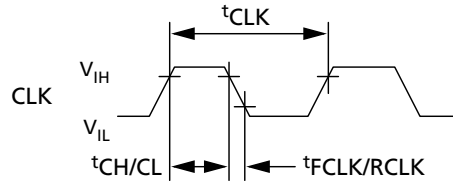


Table 37: Capacitance

Notes apply to all parameters.

| Parameter | Symbol | Signals | Min | Typ | Max | Unit | Test Condition |
|--------------------|------------------|--|-----|-----|-----|------|---------------------------|
| Input capacitance | C _{IN} | Address, CLK, CE#, OE#, ADV#, WE#, WP#, and RST# | 2 | 4 | 6 | pF | V _{IN} = 0–2.0V |
| Output capacitance | C _{OUT} | Data, WAIT | 2 | 5 | 6 | pF | V _{OUT} = 0–2.0V |

- Notes:
1. T_A = +25°C, f = 1 MHz.
 2. Sampled, not 100% tested.
 3. Silicon die capacitance only. For discrete packages, add 1pF. For stacked packages, total capacitance = 2pF + sum of silicon die capacitances.

AC Read Specifications

Table 38: AC Read Specifications (CLK-Latching, 133 MHz), $V_{CCQ} = 1.7V$ to $2.0V$

Note 1 applies to all parameters

| Parameter | Symbol | Min | Max | Unit | Notes |
|--|-----------------|------|------|------------|-------|
| Asynchronous Specifications | | | | | |
| READ cycle time | t_{AVAV} | 106 | – | ns | |
| Address to output valid | t_{AVQV} | – | 106 | ns | |
| CE# LOW to output valid | t_{ELQV} | – | 106 | ns | |
| OE# LOW to output valid | t_{GLQV} | – | 7 | ns | 2 |
| RST# HIGH to output valid | t_{PHQV} | – | 150 | ns | |
| CE# LOW to output in Low-Z | t_{ELQX} | 0 | – | ns | 3 |
| OE# LOW to output in Low-Z | t_{GLQX} | 0 | – | ns | 2, 3 |
| CE# HIGH to output in High-Z | t_{EHQZ} | – | 7 | ns | 3 |
| OE# HIGH to output in High-Z | t_{GHQZ} | – | 7 | ns | 3 |
| Output hold from first occurring address, CE#, or OE# change | t_{OH} | 0 | – | ns | 3 |
| CE# pulse width HIGH | t_{EHEL} | 7 | – | ns | |
| CE# LOW to WAIT valid | t_{ELTV} | – | 8 | ns | |
| CE# HIGH to WAIT High-Z | t_{EHTZ} | – | 7 | ns | 3 |
| OE# HIGH to WAIT valid | t_{GHTV} | – | 5.5 | ns | |
| OE# LOW to WAIT valid | t_{GLTV} | – | 5.5 | ns | |
| OE# LOW to WAIT in Low-Z | t_{GLTX} | 0 | – | ns | 3 |
| Latching Specifications | | | | | |
| Address setup to ADV# HIGH | t_{AVVH} | 5 | – | ns | |
| CE# LOW to ADV# HIGH | t_{ELVH} | 7 | – | ns | |
| ADV# LOW to output valid | t_{VLQV} | – | 106 | ns | |
| ADV# pulse width LOW | t_{VLVH} | 7 | – | ns | |
| ADV# pulse width HIGH | t_{VHVL} | 7 | – | ns | |
| Address hold from ADV# HIGH | t_{VHAX} | 5 | – | ns | |
| ADV# HIGH to OE# LOW | t_{VHGL} | 2 | – | ns | |
| RST# HIGH to ADV# HIGH | t_{PHVH} | 30 | – | ns | |
| Clock Specifications | | | | | |
| CLK frequency | f_{CLK} | – | 133 | MHz | |
| CLK period | t_{CLK} | 7.5 | – | ns | |
| CLK HIGH/LOW time | $t_{CH/CL}$ | 0.45 | 0.55 | CLK period | |
| CLK fall/rise time | $t_{FCLK/RCLK}$ | 0.3 | 1.2 | ns | |
| Synchronous Specifications | | | | | |
| Address setup to CLK HIGH | t_{AVCH} | 2 | – | ns | |
| ADV# LOW setup to CLK HIGH | t_{VLCH} | 2 | – | ns | |

Table 38: AC Read Specifications (CLK-Latching, 133 MHz), $V_{CCQ} = 1.7V$ to $2.0V$ (Continued)

Note 1 applies to all parameters

| Parameter | Symbol | Min | Max | Unit | Notes |
|--|------------|-----|-----|--------|-------|
| CE# LOW setup to CLK HIGH | t_{ELCH} | 2.5 | – | ns | |
| CLK to output valid | t_{CHQV} | – | 6 | ns | |
| Output hold from CLK HIGH | t_{CHQX} | 2 | – | ns | |
| Address hold from CLK HIGH | t_{CHAX} | 1.5 | – | ns | |
| CLK HIGH to WAIT valid | t_{CHTV} | – | 6 | ns | |
| ADV# HIGH hold from CLK | t_{CHVL} | 2 | – | ns | |
| WAIT hold from CLK | t_{CHTX} | 2 | – | ns | |
| ADV# hold from CLK HIGH | t_{CHVH} | 2 | – | ns | |
| CLK to OE# LOW | t_{CHGL} | 2 | – | ns | |
| Read access time from address latching clock | t_{ACC} | 106 | – | ns | |
| ADV# pulse width LOW for sync reads | t_{VLVH} | 1 | 2 | clocks | |
| ADV# HIGH to CLK HIGH | t_{VHCH} | 2 | – | ns | |

- Notes:
1. See Electrical Specifications – AC Characteristics and Operating Conditions for timing measurements and MAX allowable input slew rate.
 2. OE# can be delayed by up to $t_{ELQV} - t_{GLQV}$ after the CE# falling edge without impact to t_{ELQV} .
 3. Sampled, not 100% tested.

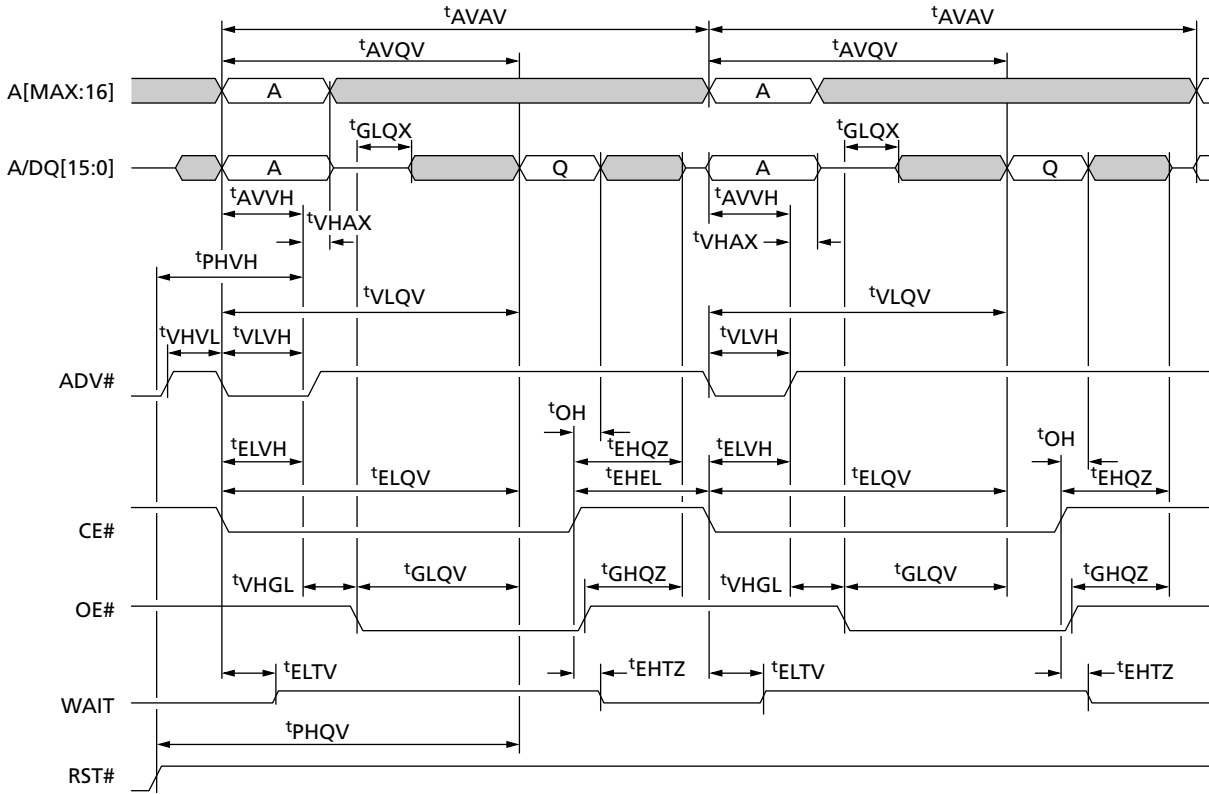
AC Read Timing

The synchronous read timing waveforms apply to both 108 and 133 MHz devices. However, devices that only support up to 108 MHz do not need to meet the following timing specifications:

- t_{CHVH}
- t_{CHGL}
- t_{ACC}
- t_{VLVH}
- t_{VHCH}

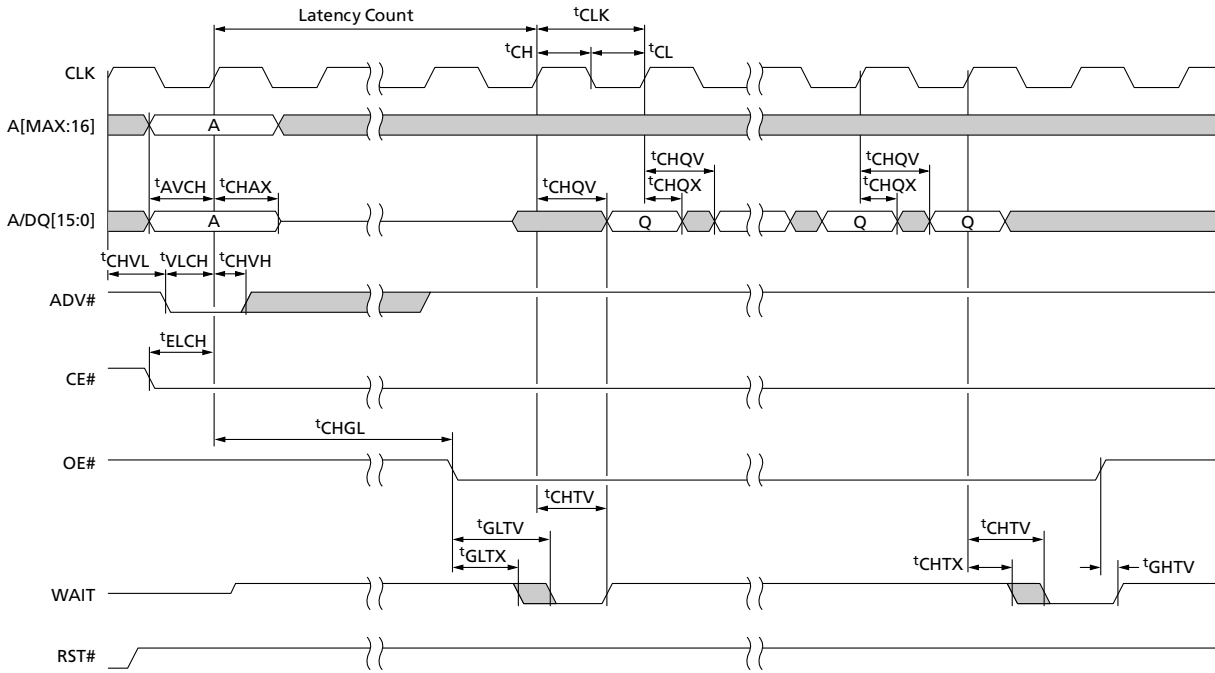
Note: The WAIT signal polarity in all the timing waveforms is low-true ($RCR10 = 0$). WAIT is shown as de-asserted with valid data ($RCR8 = 0$). WAIT is de-asserted during asynchronous reads.

Figure 16: Asynchronous Single-Word Read



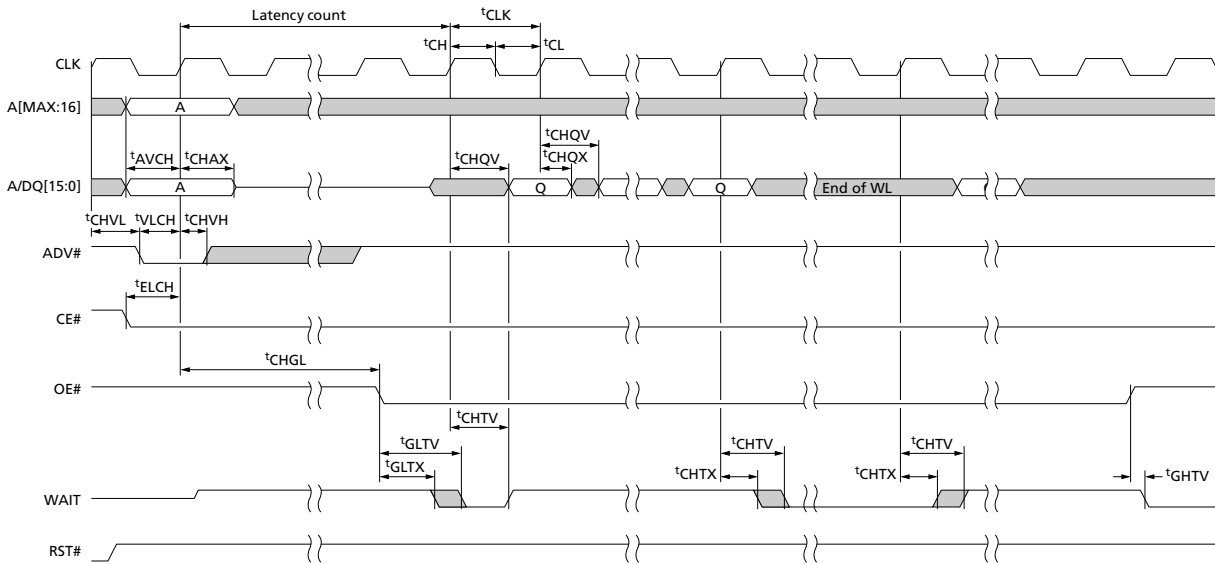
- Notes:
1. WAIT shown as active LOW (RCR[10] = 0).
 2. Back-to-back READ operations shown.
 3. CE# does not need to toggle between read cycles (i.e., t_{EHEL} need not apply).

Figure 17: Synchronous 8- or 16-Word Burst Read (A/D MUX)



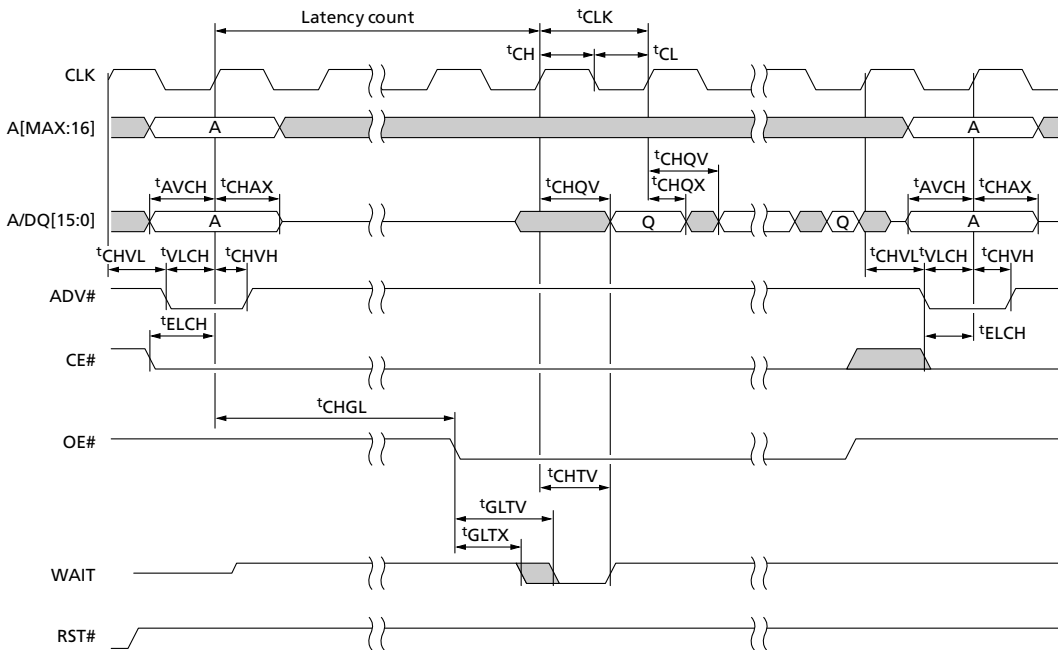
- Notes:
1. 8-word and 16-word burst are always wrap-only.
 2. WAIT shown as active LOW (RCR[10] = 0) and asserted with data (RCR[8] = 0).
 3. t_{AVQV} , t_{ELQV} , and t_{VLQV} apply to legacy-latching only.
 4. t_{ACC} and t_{VLVH} apply to clock-latching only.

Figure 18: Synchronous Continuous Misaligned Burst Read (A/D MUX)



- Notes:
1. 8-word and 16-word burst are always wrap-only.
 2. WAIT shown as active LOW (RCR[10] = 0) and asserted with data (RCR[8] = 0).
 3. t_{AVQV} , t_{ELQV} , and t_{VLQV} apply to legacy-latching only.
 4. t_{ACC} and t_{VLVH} apply to clock-latching only.

Figure 19: Synchronous Burst with Burst-Interrupt (AD-MUX)



- Notes:
1. t_{AVQV} , t_{ELQV} , and t_{VLQV} apply to legacy-latching only.
 2. t_{ACC} and t_{VLVH} apply to clock-latching only.



256Mb, 512Mb, 1Gb StrataFlash Memory AC Read Specifications

3. A burst can be interrupted by toggling CE# or ADV#.

AC Write Specifications

Table 39: AC Write Specifications

Notes 1 and 2 apply to all

| Parameter | Symbol | Min | Max | Unit | Notes |
|--|------------|-----------------|-----|------|----------|
| RST# HIGH recovery to WE# LOW | t_{PHWL} | 150 | – | ns | 3 |
| CE# setup to WE# LOW | t_{ELWL} | 0 | – | ns | 10 |
| WE# write pulse width LOW | t_{WLWH} | 40 | – | ns | 4 |
| Data setup to WE# HIGH | t_{DVWH} | 40 | – | ns | |
| CE# hold from WE# HIGH | t_{WHEH} | 0 | – | ns | |
| Data hold from WE# HIGH | t_{WHDX} | 0 | – | ns | |
| WE# pulse width HIGH | t_{WHWL} | 20 | – | ns | 5 |
| V _{PP} setup to WE# HIGH | t_{VPWH} | 200 | – | ns | 3, 7 |
| V _{PP} hold from status read | t_{QVVL} | 0 | – | ns | 3, 7 |
| WP# hold from status read | t_{QVBL} | 0 | – | ns | 3, 7 |
| WP# setup to WE# HIGH | t_{BHWH} | 200 | – | ns | 3, 7 |
| WE# HIGH to OE# LOW | t_{WHGL} | 0 | – | ns | 8 |
| ADV# LOW to WE# HIGH | t_{VLWH} | 55 | – | ns | |
| WE# HIGH to read valid | t_{WHQV} | $t_{AVQV} + 30$ | – | ns | 3, 6, 9 |
| WRITE Operation to Asynchronous Read Transition | | | | | |
| WE# HIGH to address valid | t_{WHAV} | | | | |
| Write to Synchronous Read Specification | | | | | |
| WE# HIGH to CLK HIGH @ 110 MHz | t_{WHCH} | 15 | – | ns | 3, 6, 11 |
| WE# HIGH to CE# LOW | t_{WHEL} | 9 | – | ns | 3, 6, 11 |
| WE# HIGH to ADV# LOW | t_{WHVL} | 7 | – | ns | 3, 6, 11 |
| Write Specifications with Clock Active | | | | | |
| ADV# HIGH to WE# LOW | t_{VHWL} | – | 30 | ns | 11 |
| CLK HIGH to WE# LOW | t_{CHWL} | – | 30 | ns | 11 |

- Notes:
1. Write timing characteristics during erase suspend are the same as WRITE-only operations.
 2. A WRITE operation can be terminated with either CE# or WE#.
 3. Sampled, not 100% tested.
 4. Write pulse width LOW (t_{WLWH} or t_{ELEH}) is defined from CE# or WE# LOW (whichever occurs last) to CE# or WE# HIGH (whichever occurs first). Hence, $t_{WLWH} = t_{ELEH} = t_{WLEH} = t_{ELWH}$.
 5. Write pulse width HIGH (t_{WHWL} or t_{EHEL}) is defined from CE# or WE# HIGH (whichever occurs first) to CE# or WE# LOW (whichever occurs last). Hence, $t_{WHWL} = t_{EHEL} = t_{WHEL} = t_{EHW}$.
 6. t_{WHCH} must be met when transitioning from a WRITE cycle to a synchronous burst read. In addition CE# or ADV# must toggle when WE# goes HIGH.
 7. V_{PP} and WP# must be at a valid level until erase or program success is determined.
 8. When performing a READ STATUS operation following any command that alters the status register, t_{WHGL} is 20ns.

9. Add 10ns if the WRITE operation results in an RCR or block lock status change for the subsequent READ operation to reflect this change.
10. Either t_{VHWL} or t_{CHWL} is required to meet the specification depending on the address latching mechanism; both of these specifications can be ignored if the clock is not toggling during the WRITE cycle.
11. If ADV# remains LOW after the WRITE cycle completes, a new READ cycle will start.

Figure 20: Write Timing

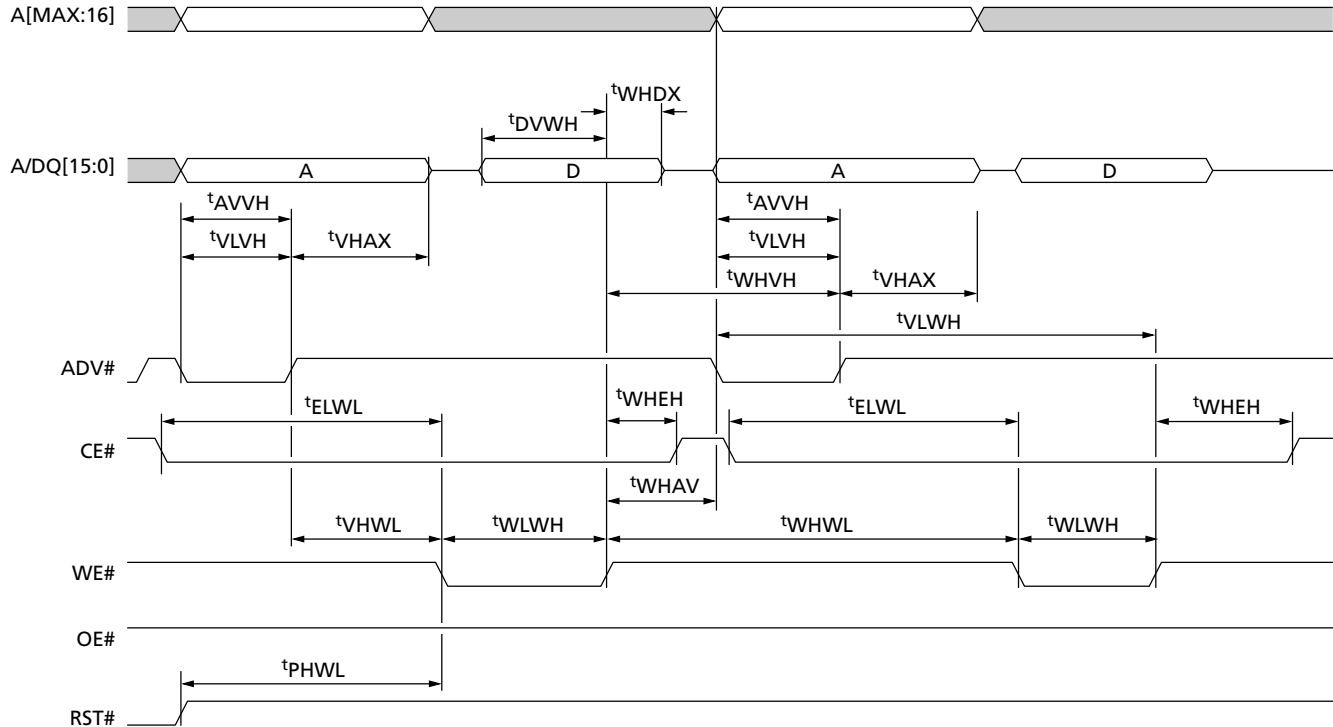


Figure 21: Write to Write (A/D-MUX)

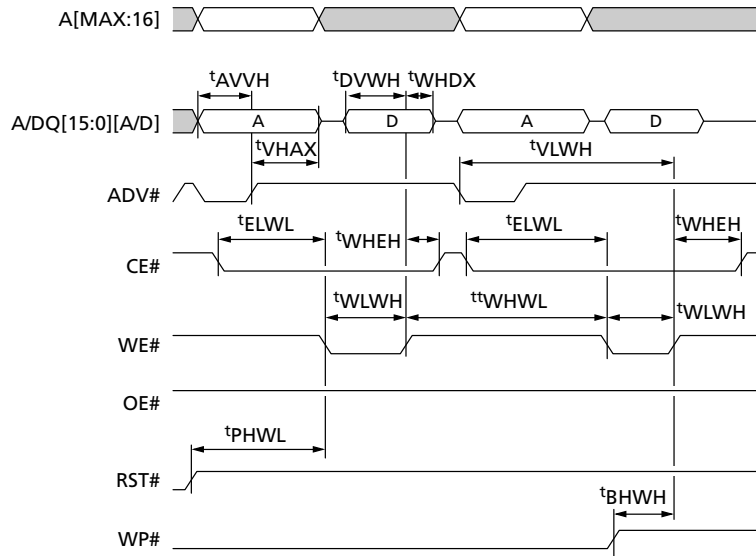


Figure 22: Async Read to Write (A/D-MUX)

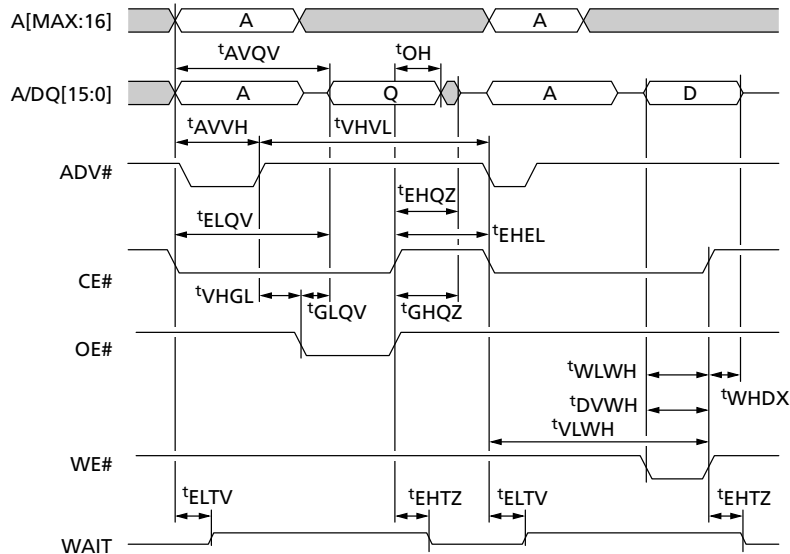


Figure 23: Write to Async Read (A/D-MUX)

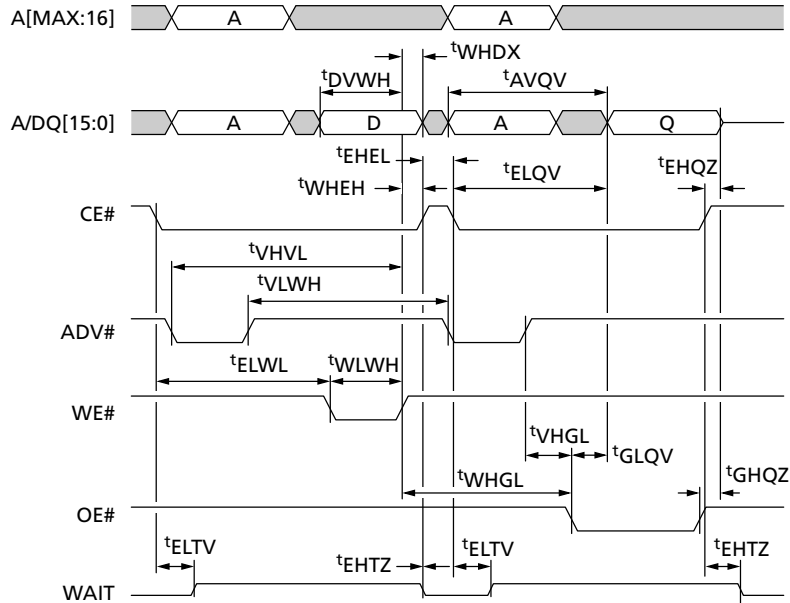


Figure 24: Sync Read to Write (A/D-MUX)

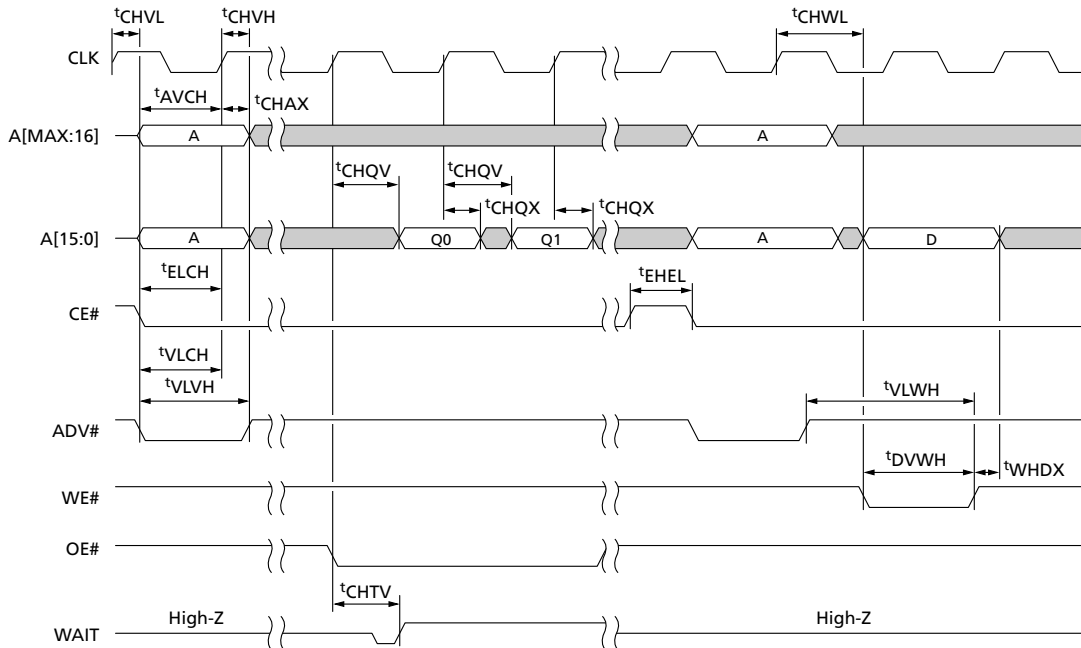
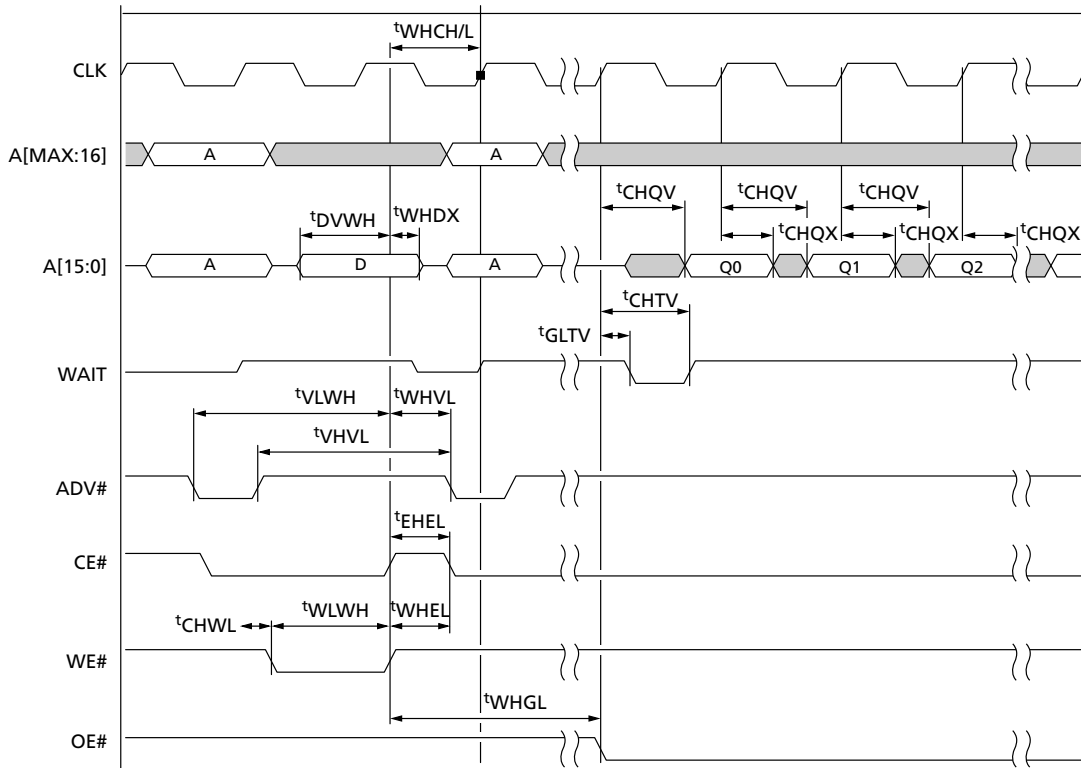


Figure 25: Write to Sync Read (A/D-MUX)





Electrical Specifications – Program/Erase Characteristics

Table 40: Program/Erase Characteristics

Note 1 applies to all

| Parameter | Symbol | V _{PPL} or V _{PPH} | | | Units | Notes | |
|---|-------------------------------|--------------------------------------|-----|------|-------|-------|---|
| | | Min | Typ | Max | | | |
| Word Programming | | | | | | | |
| Program time | Single word (first word) | ^t PROG/W | – | 115 | 230 | μs | 2 |
| | Single word (subsequent word) | | | 65 | 230 | | |
| Buffered Programming | | | | | | | |
| Program time | Single word | ^t PROG/W | – | 250 | 500 | μs | |
| | One buffer (512 words) | ^t PROG/PB | | 1.02 | 2.05 | ms | |
| Buffer Enhanced Factory Programming (BEFP) | | | | | | | |
| Program | Single word | ^t BEFP/W | | 2.0 | | μs | 3 |
| | BEFP setup | ^t BEFP/ SETUP | 5 | – | – | μs | 3 |
| Erasing and Suspending | | | | | | | |
| Erase time | 128K-word parameter | ^t ERS/MAB | – | 0.9 | 4 | s | |
| Suspend latency | Program suspend | ^t SUSP/P | – | 20 | 30 | μs | |
| | Erase suspend | ^t SUSP/E | – | 20 | 30 | μs | |
| Blank Check | | | | | | | |
| Main array block | Main array block | ^t BC/MB | – | 3.2 | – | ms | |

- Notes:
1. Typical values measured at T_A = 25°C and nominal voltages. Performance numbers are valid for all speed versions. Excludes overhead. Sampled, but not 100% tested.
 2. Conventional word programming: First and subsequent words refer to first word and subsequent words in control mode programming region.
 3. Averaged over the entire device. BEFP is not validated at V_{PPL}.

Common Flash Interface

The common Flash interface (CFI) is part of an overall specification for multiple command set and control interface descriptions. System software can parse the CFI database structure to obtain information about the device, such as block size, density, bus width, and electrical specifications. The system software determines which command set to use to properly perform a WRITE, BLOCK ERASE, or READ command, and to otherwise control the device. Information in the CFI database can be viewed by issuing the READ CFI command.

READ CFI Structure Output

The READ CFI command obtains CFI database structure information and always outputs it on the lower byte, DQ[7:0], for a word-wide (x16) Flash device. This CFI-compliant device always outputs 00h data on the upper byte (DQ[15:8]).

The numerical offset value is the address relative to the maximum bus width that the device supports, with a starting address of 010h, which is a word address for x16 devices. For example, at a starting address of 10h, a READ CFI command outputs an ASCII Q in the lower byte and 00h in the higher byte.

In the following tables, address and data are represented in hexadecimal notation. In addition, because the upper byte of word-wide devices is always 00h, the leading 00 has been dropped and only the lower byte value is shown.

Table 41: Example of CFI Output (x16 Device) as a Function of Device and Mode

| Device | Hex Offset | Hex Code | ASCII Value (DQ[15:8]) | ASCII Value (DQ[7:0]) |
|---------|------------|--------------------|------------------------|------------------------------|
| Address | 00010: | 51 | 00 | Q |
| | 00011: | 52 | 00 | R |
| | 00012: | 59 | 00 | Y |
| | 00013: | P_ID _{LO} | 00 | Primary vendor ID |
| | 00014: | P_ID _{HI} | 00 | |
| | 00015: | P _{LO} | 00 | Primary vendor table address |
| | 00016: | P _{HI} | 00 | |
| | 00017: | A_ID _{LO} | 00 | Alternate vendor ID |
| | 00018: | A_ID _{HI} | 00 | |
| | : | : | : | : |
| : | : | : | : | |

Table 42: CFI Database: Addresses and Sections

| Address | Section Name | Description |
|----------|------------------------------|--|
| 00001:Fh | Reserved | Reserved for vendor-specific information |
| 00010h | CFI ID string | Command set ID (identification) and vendor data offset |
| 0001Bh | System interface information | Timing and voltage |
| 00027h | Device geometry definition | Layout |

Table 42: CFI Database: Addresses and Sections (Continued)

| Address | Section Name | Description |
|---------|--|--|
| P | Primary Micron-specific extended query | Vendor-defined informaton specific to the primary vendor algorithm (offset 15 defines <i>P</i> which points to the primary Micron-specific extended query table) |

CFI ID String

The CFI ID string provides verification that the device supports the CFI specification. It also indicates the specification version and supported vendor-specific command sets.

Table 43: CFI ID String

| Hex Offset | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|---|---------|----------|---|
| 10h | 3 | Query unique ASCII string "QRY" | 10: | - -51 | Q |
| | | | 11: | - -52 | R |
| | | | 12: | - -59 | Y |
| 13h | 2 | Primary vendor command set and control interface ID code;16-bit ID code for vendor-specified algorithms | 13: | - -00 | Primary vendor ID number |
| | | | 14: | - -02 | |
| 15h | 2 | Extended query table primary algorithm address | 15: | - -0A | Primary vendor table address, primary algorithm |
| | | | 16: | - -01 | |
| 17h | 2 | Alternate vendor command set and control interface ID code; 0000h indicates no second vendor-specified algorithm exists | 17: | - -00 | Alternate vendor ID number |
| | | | 18: | - -00 | |
| 19h | 2 | Secondary algorithm extended query table address; 0000h indicates none exists | 19: | - -00 | Primary vendor table address, secondary algorithm |
| | | | 1A: | - -00 | |

System Interface Information

Table 44: System Interface Information

| Hex Offset | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|--|---------|----------|-----------------------|
| 1Bh | 1 | V _{CC} logic supply minimum program/erase voltage bits 0–3 BCD 100mV bits 4–7 BCD volts | 1Bh | - -17 | 1.7V |
| 1Ch | 1 | V _{CC} logic supply maximum program/erase voltage bits 0–3 BCD 100mV bits 4– 7 BCD volts | 1Ch | - -20 | 2.0V |
| 1Dh | 1 | V _{PP} [programming] supply minimum program/erase voltage bits 0–3 BCD 100mV bits 4–7 hex volts | 1Dh | - -85 | 8.5V |

Table 44: System Interface Information (Continued)

| Hex Offset | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|--|---------|---|--|
| 1Eh | 1 | V _{PP} [programming] supply maximum program/erase voltage bits 0–3 BCD 100mV bits 4–7 hex volts | 1Eh | --95 | 9.5V |
| 1Fh | 1 | n such that typical single word program timeout = $2^n\mu\text{s}$ | 1Fh | --06 | 64 μs |
| 20h | 1 | n such that typical full buffer write timeout = $2^n\mu\text{s}$ | 20h | --0B (1024Mb) --0A (256Mb, 512Mb) | 2048 μs (1024Mb) 1023 μs (256Mb, 512Mb) |
| 21h | 1 | n such that typical block erase timeout = 2^nms | 21h | --0A | 1s |
| 22h | 1 | n such that typical full chip erase timeout = 2^nms | 22h | --00 | NA |
| 23h | 1 | n such that maximum word program timeout = 2^n times typical | 23h | --02 | 256 μs |
| 24h | 1 | n such that maximum buffer write timeout = 2^n times typical | 24h | --02 (256Mb, 512Mb) --01 (1024Mb) | 8192 μs (256Mb, 512Mb) 4096 μs (1024Mb) |
| 25h | 1 | n such that maximum block erase timeout = 2^n times typical | 25h | --02 | 4s |
| 26h | 1 | n such that maximum chip erase timeout = 2^n times typical | 26h | --00 | NA |

Device Geometry Definition

Table 45: Device Geometry

| Hex Offset | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|--|---------|----------|-----------------------|
| 27h | 1 | n such that device size in bytes = 2^n . | 27: | | See table below |
| 28h | 2 | Flash device interface code assignment: n such that $n + 1$ specifies the bit field that represents the device width capabilities as described here: bit 0: x8 bit 1: x16 bit 2: x32 bit 3: x64 bits 4–7: – bits 8–15: – | 28: | --01 | x16 |
| | | | 29: | --00 | |

Table 45: Device Geometry (Continued)

| Hex Offset | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------|--------|---|------------|----------|-----------------------|
| 2Ah | 2 | n such that maximum number of bytes in write buffer = 2^n | 2Ah | - -0A | 1024 |
| | | | 2Bh | - -00 | |
| 2Ch | 1 | Number of erase block regions (x) within the device: x = 0 indicates no erase blocking; the device erases in bulk x specifies the number of device regions with one or more contiguous, same-size erase blocks Symmetrically blocked partitions have one blocking region | 2Ch | | See table below |
| 2Dh | 4 | Erase block region 1 information: bits 0–15 = y, y + 1 = number of identical-size erase blocks bits 16–31 = z, region erase block(s) size are z x 256 bytes | 2D: 30: | | See table below |

Note: 1. See the bit field table.

Table 46: Block Region Map Information

| Address | 256Mb | | 512Mb | | 1Gb | |
|---------|--------|-----|--------|-----|--------|-----|
| | Bottom | Top | Bottom | Top | Bottom | Top |
| 27: | - -19 | -- | - -1A | -- | - -1B | -- |
| 28: | - -01 | -- | - -01 | -- | - -01 | -- |
| 29: | - -00 | -- | - -00 | -- | - -00 | -- |
| 2A: | - -0A | -- | - -0A | -- | - -0A | -- |
| 2B: | - -00 | -- | - -00 | -- | - -00 | -- |
| 2C: | - -01 | -- | - -01 | -- | - -01 | -- |
| 2D: | - -7F | -- | - -FF | -- | - -FF | -- |
| 2E: | - -00 | -- | - -00 | -- | - -01 | -- |
| 2F: | - -00 | -- | - -00 | -- | - -00 | -- |
| 30: | - -04 | -- | - -04 | -- | - -04 | -- |

Primary Micron-Specific Extended Query

Table 47: Primary Micron-Specific Extended Query

| Hex Offset P = 10Ah | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) | |
|--------------------------------------|--------|---|---|------------------------|--------------------------|------------------------------|
| (P+0)h (P+1)h (P+2)h | 3 | Primary extended query table, unique ASCII string: PRI | 10A: | - -50 | P | |
| | | | 10B: | - -52 | R | |
| | | | 10C: | - -49 | I | |
| (P+3)h | 1 | Major version number, ASCII | 10D: | - -31 | 1 | |
| (P+4)h | 1 | Minor version number, ASCII | 10E: | - -34 | 4 | |
| (P+5)h (P+6)h (P+7)h (P+8)h | 4 | Optional feature and command support (1 = yes; 0 = no) Bits 10–31 are reserved; undefined bits are 0 If bit 31 = 1, then another 31-bit field of optional features follows at the end of the bit 30 field | 10F: | - -66 | – | |
| | | | 110: | - -07 | – | |
| | | | 111: | - -00 | – | |
| | | | 112: | - -00 | – | |
| | | | Bit 0: Chip erase supported | Bit 0 = 0 | | No |
| | | | Bit 1: Suspend erase supported | Bit 1 = 1 | | Yes |
| | | | Bit 2: Suspend program supported | Bit 2 = 1 | | Yes |
| | | | Bit 3: Legacy lock/unlock supported | Bit 3 = 0 | | No |
| | | | Bit 4: Queued erase supported | Bit 4 = 0 | | No |
| | | | Bit 5: Instant individual block locking supported | Bit 5 = 1 | | Yes |
| | | | Bit 6: OTP bits supported | Bit 6 = 1 | | Yes |
| | | | Bit 7: Page mode read supported | Bit 7 = 0 Bit 7 = 1 | | No: A/D MUX Yes: Reserved |
| | | | Bit 8: Synchronous read supported | Bit 8 = 1 | | Yes |
| | | | Bit 9: Simultaneous operations supported | Bit 9 = 0 | | No |
| | | | Bit 10: Reserved | Bit 10 = 0 | | No |
| | | | Bit 30: CFI links to follow | Bit 30 = 0 | | No |
| | | Bit 31: another optional features field to follow. | Bit 31 = 0 | | No | |
| (P+9)h | 1 | Supported functions after suspend: read array, status, query Other supported options: Bits 1–7 reserved; undefined bits are 0 | 113: | - -01 | – | |
| | | Bit 0: Program supported after erase suspend | Bit 0 = 1 | | Yes | |
| (P+A)h (P+B)h | 2 | Block status register mask: Bits 2 – 3 and 6 - 15 are reserved; undefined bits are 0 | 114: | - -33 | – | |
| | | | 115: | – | | |
| | | Bit 0: Block lock bit status register active | Bit 0 = 1 | | Yes | |
| | | Bit 1: Block lock-down bit status active | Bit 1 = 1 | | Yes | |
| (P+C)h | 1 | V _{CC} logic supply highest performance program/ erase voltage bits 0–3 BCD 100mV Bits 4–7 BCD value in volts | 116: | - -18 | 1.8V | |

Table 47: Primary Micron-Specific Extended Query (Continued)

| Hex Offset P = 10Ah | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------------------|--------|--|---------|----------|--------------------------|
| (P+D)h | 1 | V _{PP} optimum program/erase voltage Bits 0–3 BCD 100mV Bits 4–7 hex value in volts | 117: | --90 | 9.0V |

Table 48: One Time Programmable (OTP) Space Information

| Hex Offset P = 10Ah | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|--|--------|---|---------|----------|--------------------------|
| (P+E)h | 1 | Number of OTP block fields in JEDEC ID space. 00h indicates that 256 OTP fields are available | 118: | --02 | 2 |
| (P+F)h (P+10)h (P+11)h (P+12)h | 4 | OTP Field 1: This field describes user-available OTP bytes. Some are preprogrammed with device-unique serial numbers. Others are user-programmable. Bits 0–15 point to the OTP lock byte (the first byte). The following bytes are factory preprogrammed and user-programmable: Bits 0–7 = lock/bytes JEDEC plane physical low address. Bits 8–15 = lock/bytes JEDEC plane physical high address. Bits 16–23 = n where 2^n equals factory-preprogrammed bytes. Bits 24–31 = n where 2^n equals user-programmable bytes. | 119: | --80 | 80h |
| | | | 11A: | --00 | 00h |
| | | | 1B: | --03 | 8 byte |
| | | | 11C: | --03 | 8 byte |
| (P+13)h (P+14)h (P+15)h (P+16)h | 4 | Protection Field 2: Bits 0–31 point to the protection register physical lock word address in the JEDEC plane. The bytes that follow are factory or user-programmable. | 11D: | --89 | 89h |
| | | | 11E: | --00 | 00h |
| | | | 11F: | --00 | 00h |
| | | | 120: | --00 | 00h |
| (P+17)h (P+18)h (P+19)h | 3 | Bits 32–39 = n where n equals factory-programmed groups (low byte). Bits 40–47 = n where n equals factory-programmed groups (high byte). Bits 48–55 = n where $2n$ equals factory-programmed bytes/groups. | 121: | --00 | 0 |
| | | | 122: | --00 | 0 |
| | | | 123: | --00 | 0 |
| (P+1A)h (P+1B)h (P+1C)h | 3 | Bits 56–63 = n where n equals user-programmed groups (low byte). Bits 64–71 = n where n equals user-programmed groups (high byte). Bits 72–79 = n where n equals user-programmable bytes/groups. | 124: | --10 | 16 |
| | | | 125: | --00 | 0 |
| | | | 126: | --04 | 16 |

Table 49: Burst Read Informaton

| Hex Offset P = 10Ah | Length | Description | Address | Hex Code | ASCII Value (DQ[7:0]) |
|------------------------|--------|--|---------|----------|--------------------------|
| (P+1D)h | 1 | Page mode read capability: Bits 7–0 = n where 2 ⁿ hex value represents the number of read page bytes. See offset 28h for device word width to determine page mode data output width. 00h indicates no read page buffer. | 127: | --00 | 0 |
| (P+1E)h | 1 | Number of synchronous mode read configuration fields that follow. 00h indicates no burst capability. | 128: | --03 | 3 |
| (P+1F)h | 1 | Synchronous mode read capability configuration 1: Bits 3–7 = reserved. Bits 0–2 = n where 2 ⁿ⁺¹ hex value represents the maximum number of continuous synchronous reads when the device is configured for its maximum word width. A value of 07h indicates that the device is capable of continuous linear bursts that will output data until the internal burst counter reaches the end of the device’s burstable address space. This fields’s 3-bit value can be written directly to the RCR bits 0–2 if the device is configured for its maximum word width. See offset 28h for word width to determine the burst data output width. | 129: | --02 | 8 |
| (P+20)h | 1 | Synchronous mode read capability configuration 2. | 12A: | --03 | 16 |
| (P+21)h | 1 | Synchronous mode read capability configuration 3. | 12B: | --07 | Cont |

Table 50: Partition and Block Erase Region Information

| Hex Offset P = 10Ah | | Description Optional Features and Commands | Length | Address | |
|------------------------|---------|---|--------|---------|------|
| Bottom | Top | | | Bottom | Top |
| (P+22)h | (P+22)h | Number of device hardware partition regions within the device: x = 0: A single hardware partition device (no fields follow). x specifies the number of device partition regions containing one or more contiguous erase block regions | 1 | 12C: | 12C: |

Table 51: Partition Region 1 Information: Top and Bottom Offset/Address

| Hex Offset P = 10Ah | | Description Optional Features and Commands | Length | Address | |
|--|--|--|--------|---------|------|
| Bottom | Top | | | Bottom | Top |
| (P+23)h | (P+23)h | Data size of this Partition Region information field: (number of addressable locations, including this field). | 2 | 12D: | 12D: |
| (P+24)h | (P+24)h | | | 12E: | 12E: |
| (P+25)h (P+26)h | (P+25)h (P+26)h | Number of identical partitions within the partition region. | 2 | 12F: | 12F: |
| | | | | 130: | 130: |
| (P+27)h | (P+27)h | Number of PROGRAM or ERASE operations allowed in a partition: Bits 0–3 = number of simultaneous PROGRAM operations. Bits 4–7 = number of simultaneous ERASE operations. | 1 | 131: | 131: |
| (P+28)h | (P+28)h | Simultaneous PROGRAM or ERASE operations allowed in other partitions while a partition in this region is in program mode: Bits 0–3 = number of simultaneous program operations. Bits 4–7 = number of simultaneous ERASE operations. | 1 | 132: | 132: |
| (P+29)h | (P+29)h | Simultaneous PROGRAM or ERASE operations allowed in other partitions while a partition in this region is in erase mode: Bits 0–3 = number of simultaneous PROGRAM operations. Bits 4–7 = number of simultaneous ERASE operations. | 1 | 133: | 133: |
| (P+2A)h | (P+2A)h | Types of erase block regions in this partition region: x = 0: no erase blocking; the partition region erases in bulk. x = number of erase block regions with contiguous, same-size erase blocks. Symmetrically blocked partitions have one blocking region. Partition size = (type 1 blocks) x (type 1 block sizes) + (type 2 blocks) x (type 2 block sizes) +...+ (type n blocks) x (type n block sizes). | 1 | 134: | 134: |
| (P+2B)h (P+2C)h (P+2D)h (P+2E)h | (P+2B)h (P+2C)h (P+2D)h (P+2E)h | Partition region 1 (erase block type 1) information: Bits 0–15 = y, y+1 = number of identical-sized erase blocks in a partition. Bits 16–31 = z, where region erase block(s) size is z x 256 bytes. | 4 | 135: | 135: |
| | | | | 136: | 136: |
| | | | | 137: | 137: |
| | | | | 138: | 138: |
| (P+2F)h (P+30)h | (P+2F)h (P+30)h | Partition 1 (erase block type 1): Minimum block erase cycles x 1000 | 2 | 139: | 139: |
| | | | | 13A: | 13A: |

Table 51: Partition Region 1 Information: Top and Bottom Offset/Address (Continued)

| Hex Offset P = 10Ah | | Description Optional Features and Commands | Length | Address | |
|------------------------|---------|--|--------|---------|------|
| Bottom | Top | | | Bottom | Top |
| (P+31)h | (P+31)h | Partition 1 (erase block type 1) bits per cell; internal EDAC: Bits 0–3 = bits per cell in erase region Bit 4 = internal EDAC used (1 = yes, 0 = no) Bits 5–7 = reserved for future use | 1 | 13B: | 13B: |
| (P+32)h | (P+32)h | Partition 1 (erase block type 1) page mode and synchronous mode capabilities: Bits 0 = page mode host reads permitted (1 = yes, 0 = no) Bit 1 = synchronous host reads permitted (1 = yes, 0 = no) Bit 2 = synchronous host writes permitted (1 = yes, 0 = no) Bits 3–7 = reserved for future use | 1 | 13C: | 13C: |
| (P+33)h | (P+33)h | Partition 1 (Erase Block Type 1) programming region information: Bits 0 - 7 = x, 2 ^x : programming region aligned size (bytes) Bits 8 - 14 = reserved for future use Bit 15 = legacy flash operation; ignore 0:7 Bit 16 - 23 = y: control mode valid size (bytes) Bit 24 - 31 = reserved for future use Bit 32 - 39 = z: control mode invalid size (bytes) Bit 40 - 46 = reserved for future use Bit 47 = legacy flash operation (ignore 23:16 and 39:32) | 6 | 13D: | 13D: |
| (P+34)h | (P+34)h | | | 13E: | 13E: |
| (P+35)h | (P+35)h | | | 13F: | 13F: |
| (P+36)h | (P+36)h | | | 140: | 140: |
| (P+37)h | (P+37)h | | | 141: | 141: |
| (P+38)h | (P+38)h | | | 142: | 142: |



Table 52: Partition and Erase Block Map Information

| Address | 256Mb | | 512Mb | | 1Gb | |
|---------|--------|-----|--------|-----|--------|-----|
| | Bottom | Top | Bottom | Top | Bottom | Top |
| 12C: | --01 | -- | --01 | -- | --01 | -- |
| 12D: | --16 | -- | --16 | -- | --16 | -- |
| 12E: | --00 | -- | --00 | -- | --00 | -- |
| 12F: | --08 | -- | --08 | -- | --08 | -- |
| 130: | --00 | -- | --00 | -- | --00 | -- |
| 131: | --11 | -- | --11 | -- | --11 | -- |
| 132: | --00 | -- | --00 | -- | --00 | -- |
| 133: | --00 | -- | --00 | -- | --00 | -- |
| 134: | --01 | -- | --01 | -- | --01 | -- |
| 135: | --0F | -- | --1F | -- | --3F | -- |
| 136: | --00 | -- | --00 | -- | --00 | -- |
| 137: | --00 | -- | --00 | -- | --00 | -- |
| 138: | --04 | -- | --04 | -- | --04 | -- |
| 139: | --64 | -- | --64 | -- | --64 | -- |
| 13A: | --00 | -- | --00 | -- | --00 | -- |
| 13B: | --12 | -- | --12 | -- | --12 | -- |
| 13C: | --02 | -- | --02 | -- | --02 | -- |
| 13D: | --0A | -- | --0A | -- | --0A | -- |
| 13E: | --00 | -- | --00 | -- | --00 | -- |
| 13F: | --10 | -- | --10 | -- | --10 | -- |
| 140: | --00 | -- | --00 | -- | --00 | -- |
| 141: | --10 | -- | --10 | -- | --10 | -- |
| 142: | --00 | -- | --00 | -- | --00 | -- |

Program Flowcharts

Figure 26: Word Program Flow Chart

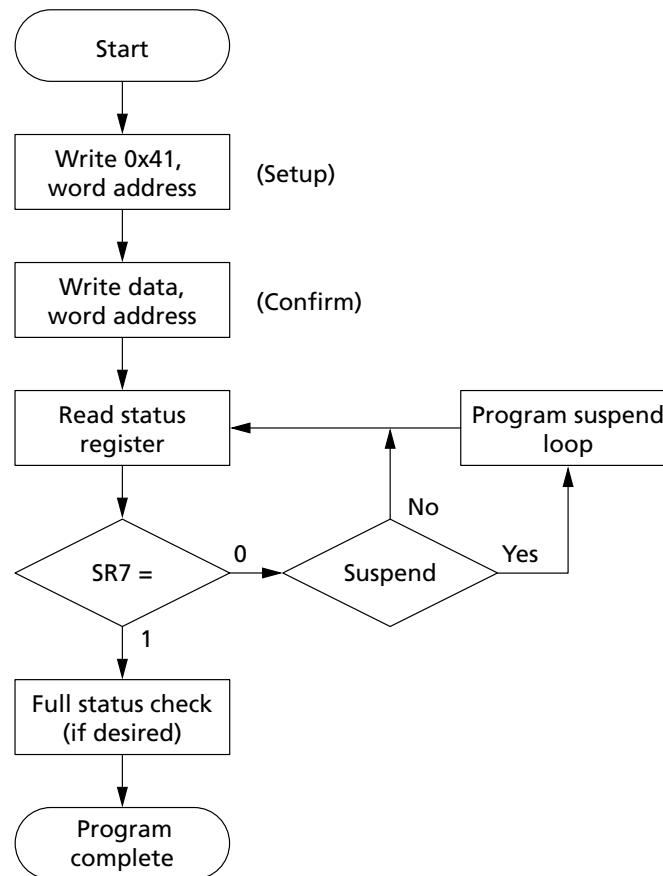


Table 53: Word Program Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|---------------|--|-------|
| WRITE | PROGRAM SETUP | Data = 0x41 Address = Location to program | 1 |
| WRITE | DATA | Data = Data to program Address = Location to program | – |
| READ | None | Status register data | – |
| Idle | None | Check SR7 1 = Write state machine ready 0 = Write state machine busy | 2, 3 |

- Notes:
1. Repeat for subsequent word PROGRAM operations.
 2. Full status register check can be done after each program or after a sequence of PROGRAM operations.
 3. Write 0xFF after the last operation to set to the read array state.

Figure 27: Word Program Full Status Check Flow Chart

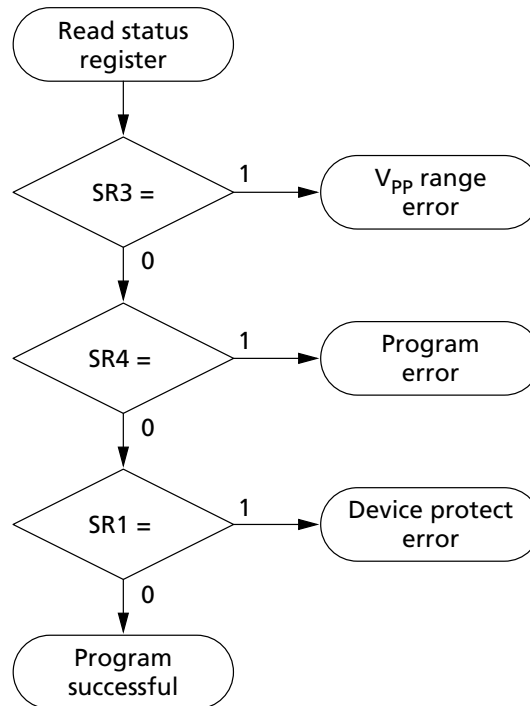


Table 54: Word Program Full Status Check Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|---------|--|-------|
| Idle | None | Check SR3 1 = V _{pp} error | 1, 2 |
| Idle | None | Check SR4 1 = Data program error | 1 |
| Idle | None | Check SR1 1 = Block locked; operation aborted | 1 |

- Notes:
1. If an error is detected, clear the status register before continuing operations. Only the CLEAR STATUS REGISTER command clears the status register error bits.
 2. SR3 MUST be cleared before the write state machine will support further program attempts.

Figure 28: Program Suspend/Resume Flow Chart

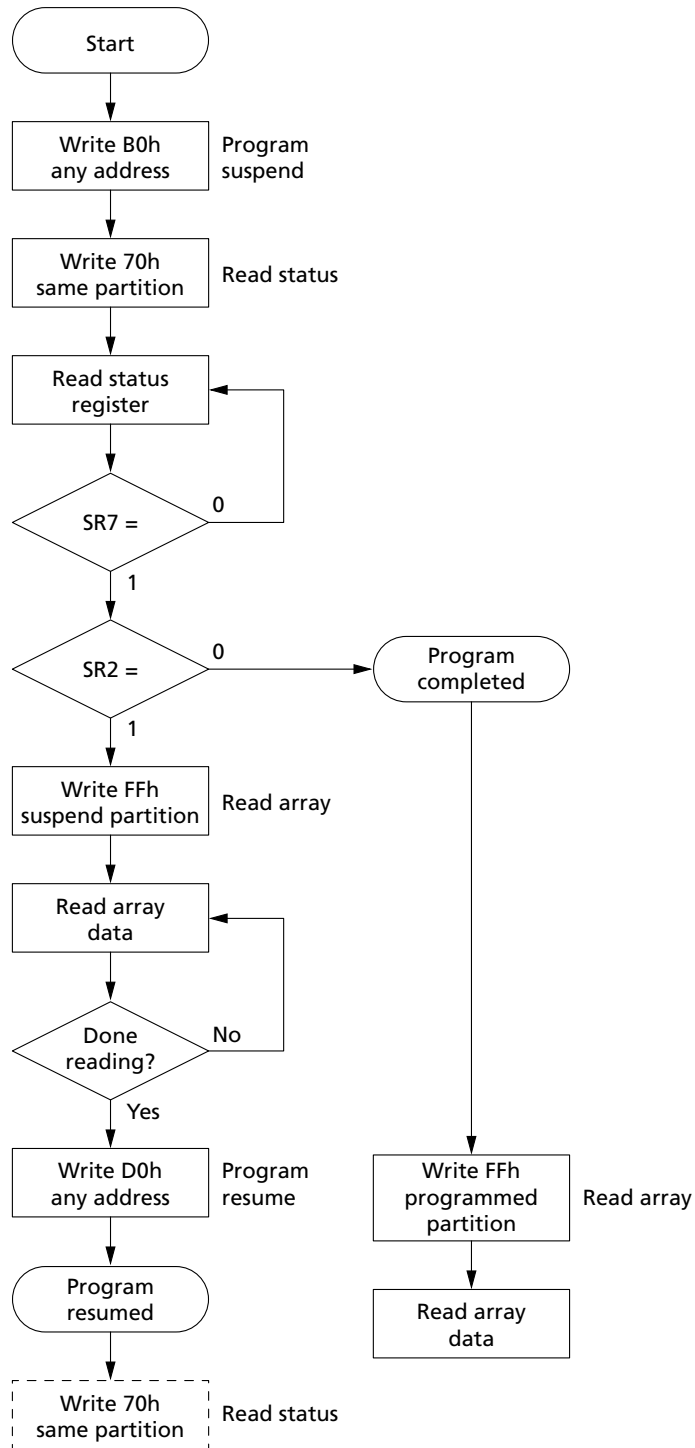


Table 55: Program Suspend/Resume Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|-----------------|--|-------|
| WRITE | PROGRAM SUSPEND | Data = B0h Address = Block to suspend | – |
| WRITE | READ STATUS | Data = 70h Address = Same partition | – |
| READ | | Status register data Address = Suspended block | – |
| Standby | | Check SR7 1 = Write state machine ready 0 = Write state machine busy | – |
| Standby | | Check SR2 1 = Program suspended 0 = Program completed | – |
| WRITE | READ ARRAY | Data = FFh Address = Any address within the suspended partition | – |
| READ | | Read array data from block other than the one being program- med | – |
| WRITE | PROGRAM RESUME | Data = D0h Address = Suspended block | – |
| WRITE | READ STATUS | Return partition to status mode: Data = 70h Address = Same partition | 1 |

Note: 1. Applicable when the suspended partition was placed in read array mode.

Figure 29: Buffer Programming Flow Chart

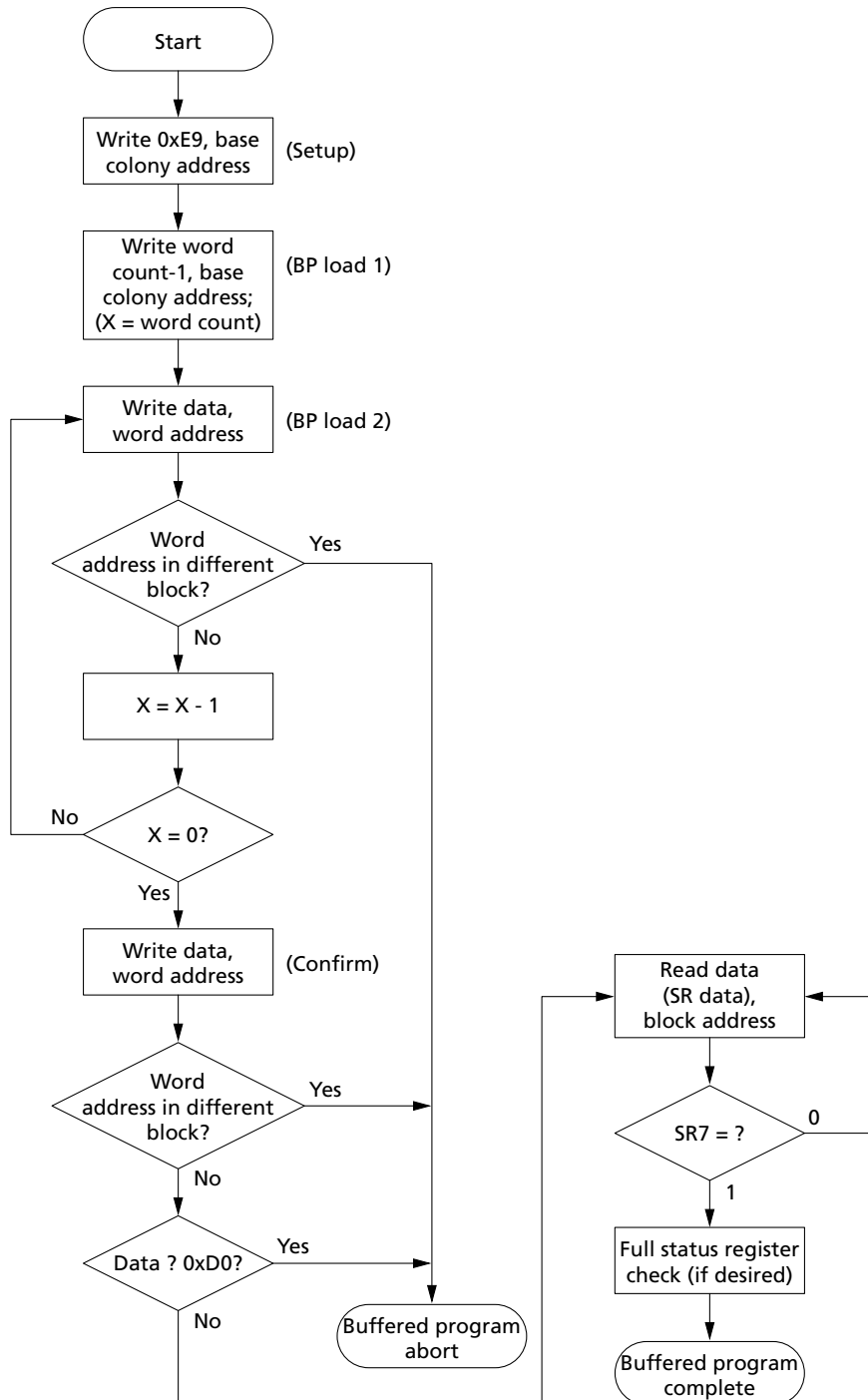


Table 56: Buffer Programming Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|--------------------------|--|-------|
| WRITE | BUFFERED PROGRAM SETUP | Data = 0xE9 Addr = Colony base address | – |
| WRITE | BUFFERED PROGRAM LOAD 1 | Data = word count -1 Address = Block address | 1 |
| WRITE | BUFFERED PROGRAM LOAD 2 | Data = Data to be programmed Address = Word address | 2, 3 |
| WRITE | BUFFERED PROGRAM CONFIRM | Data = 0xD0 Address = Address within block | 4, 5 |
| READ | None | Status register Data Address = Block address | – |

- Notes:
1. D[8:0] is loaded as word count-1.
 2. Repeat BUFFERED PROGRAM LOAD 2 until the word count is achieved. (Load up to 512 words.)
 3. The command sequence aborts if the address of the BUFFERED PROGRAM LOAD 2 cycle is in a different block from the address of the BUFFERED PROGRAM SETUP cycle.
 4. The command sequence aborts if the address of the BUFFERED PROGRAM CONFIRM cycle is in a different block from the address of the BUFFERED PROGRAM SETUP cycle. Also, an abort will occur if the data of the BUFFERED PROGRAM CONFIRM cycle data is not 0xD0.
 5. The read mode changes to status read on the BUFFERED PROGRAM CONFIRM command.

Figure 30: Buffered Enhanced Factory Programming (BEFP) Flow Chart

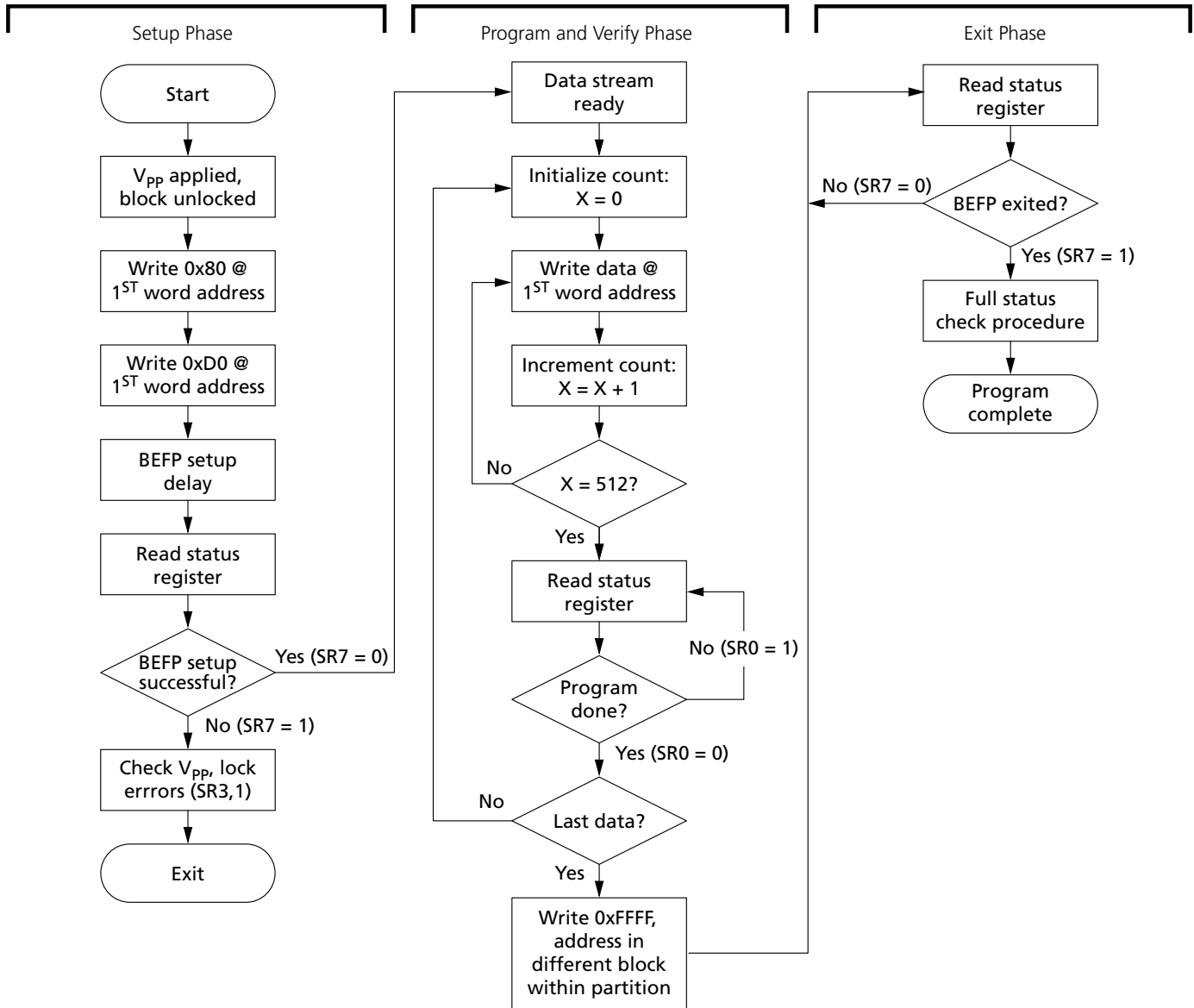


Table 57: Buffered Enhanced Factory Programming (BEFP) Procedure

| Bus Operation | Action | Comments | Notes |
|--------------------|-----------------|---|-------|
| Setup Phase | | | |
| WRITE | Unlock block | V _{PPH} applied to V _{PP} | |
| WRITE | BEFP setup | Data = 0x80 @ first word address | 1 |
| WRITE | BEFP confirm | Data = 0xD0 @ first word address | |
| READ | Status register | Data = Status register data Address = First word address | |

Table 57: Buffered Enhanced Factory Programming (BEFP) Procedure (Continued)

| Bus Operation | Action | Comments | Notes |
|---------------------------------|-------------------------------|---|-------|
| Standby | BEFP setup done? | Check SR7: 0 = BEFP ready 1 = BEFP not ready | |
| Standby | Error condition check | If SR7 is set, check: SR3 set = V_{PP} error SR1 set = Locked block | |
| Program and Verify Phase | | | |
| READ | Status register | Data = Status register data Address = First word address | |
| Standby | Data stream ready? | Check SR0: 0 = Ready for data 1 = Not ready for data | |
| Standby | Initialize count | $X = 0$ | |
| WRITE | Load buffer | Data = Data to program Address = First word address ² | |
| Standby | Increment count | $X = X + 1$ | |
| Standby | Buffer full? | $X = 512?$ Yes = Read SR0 No = Load next data word | |
| READ | Status register | Data = Status register data Address = First word address | |
| Standby | Program done? | Check SR0: 0 = Program done 1 = Program in progress | |
| Standby | Last data? | No = Fill buffer again Yes = Exit | |
| WRITE | Exit program and verify phase | Data = 0xFFFF @ address not in current block | |
| Exit Phase | | | |
| READ | Status register | Data = Status register data Address = First word address | 2, 3 |
| Standby | Check exit status | Check SR7: 0 = Exit not completed 1 = Exit completed | |

- Notes:
1. Repeat for subsequent blocks.
 2. After BEFP exit, a full status register check can determine if any program error occurred.
 3. See the Word Program Full Status Register Check Procedure flowchart.
 4. Write 0xFF to enter read array state.

Erase Flowcharts

Figure 31: Block Erase Flowchart

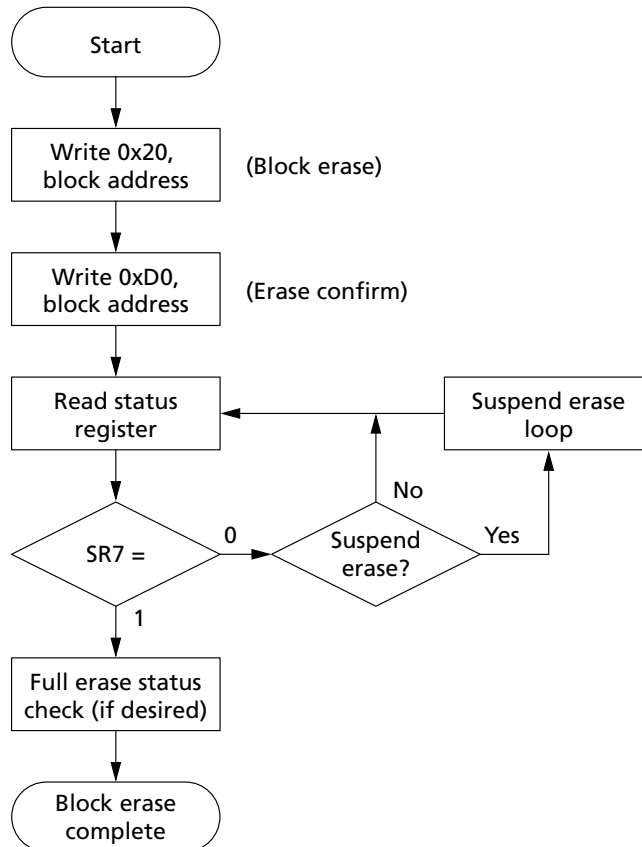


Table 58: Block Erase Procedure

| Bus Operation | Command | Comments |
|---------------|-------------------|--|
| WRITE | BLOCK ERASE SETUP | Data = 0x20 Address = Block to be erased |
| WRITE | ERASE CONFIRM | Data = 0xD0 Address = Block to be erased |
| READ | None | Status register data |
| Idle | None | Check SR7 1 = Write state machine ready 0 = Write state machine busy |

- Notes:
1. Repeat for subsequent block erasures.
 2. Full status register check can be done after each block erase or after a sequence of block erasures.
 3. Write 0xFF after the last operation to enter read array mode.

Figure 32: Block Erase Full Status Check Flow Chart

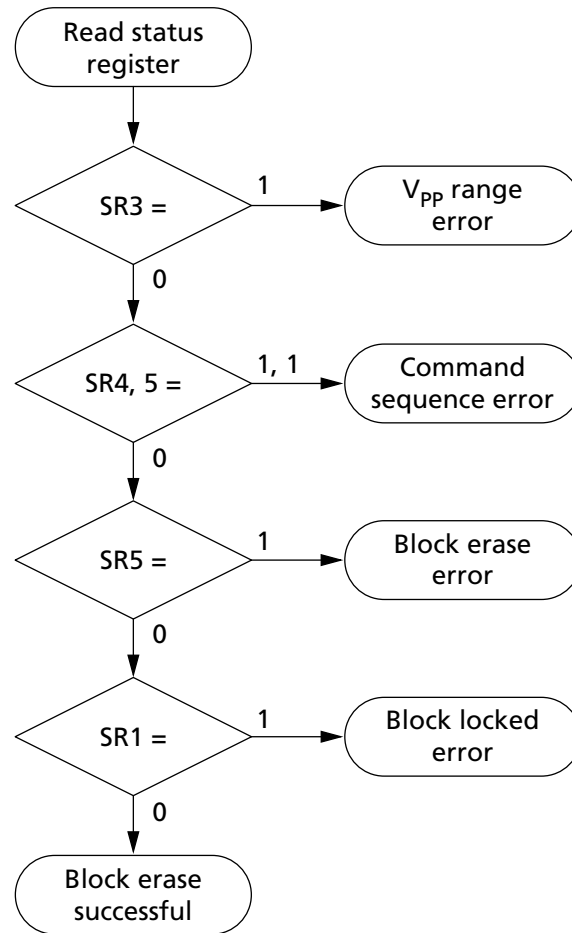


Table 59: Block Erase Full Status Check Procedure

Notes apply to entire table.

| Bus Operation | Command | Comments | Notes |
|---------------|---------|---|-------|
| Idle | None | Check SR3 1 = V _{pp} range error | – |
| Idle | None | Check SR[4, 5] Both 1 = Command sequence error | – |
| Idle | None | Check SR5 1 = Block erase error | – |
| Idle | None | Check SR1 1 = Attempted erase of locked block; erase aborted | – |

- Notes:
1. Only the CLEAR STATUS REGISTER command clears the SR[1, 3, 4, 5].
 2. If an error is detected, clear the status register before attempting an erase retry or other error recovery.

Figure 33: Erase Suspend/Resume Flow Chart

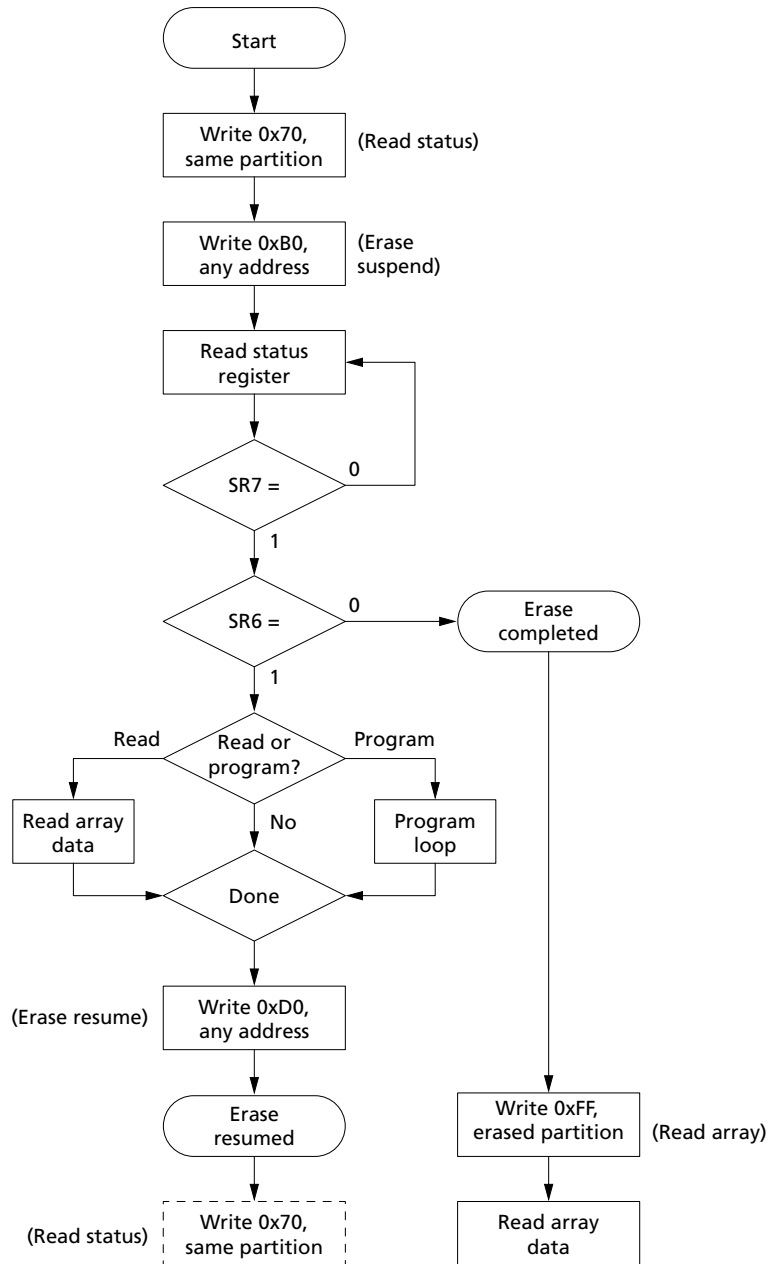


Table 60: Erase Suspend/Resume Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|----------------------|--|-------|
| WRITE | READ STATUS | Data = 0x70 Address = Any partition address | – |
| WRITE | ERASE SUSPEND | Data = 0xB0 Address = Same partition address as above | – |
| READ | None | Status register data Address = Same partition | – |
| Idle | None | Check SR7 1 = Write state machine ready 0 = Write state machine busy | – |
| Idle | None | Check SR6 1 = Erase suspended 0 = Erase completed | – |
| WRITE | Any READ or PROGRAM | Data = Command for desired operation Address = Any address within the suspended partition | – |
| READ or WRITE | None | Read array or program data from/to block other than the one being erased | – |
| WRITE | PROGRAM RESUME | Data = 0xD0 Address = Any address | – |
| WRITE | READ STATUS REGISTER | Return partition to status mode: Data = 0x70 Address = Same partition | 1 |

Note: 1. Applicable when the suspended partition was placed in read array mode or a program loop.

Block Lock and Protection Flowcharts

Figure 34: Block Lock Operations Flow Chart

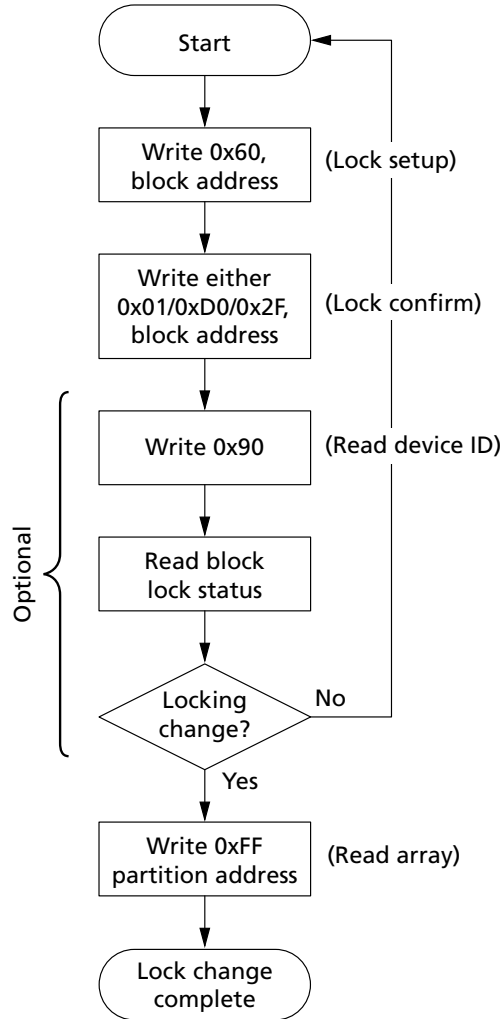


Table 61: Block Lock Operations Procedure

| Bus Operation | Command | Comments |
|------------------|------------------------------------|---|
| WRITE | LOCK SETUP | Data = 0x60 Address = Block to lock/unlock/lock-down |
| WRITE | LOCK, UNLOCK, or LOCK-DOWN CONFIRM | Data = 0x01 (BLOCK LOCK) Data = 0xD0 (BLOCK UNLOCK) Data = 0x2F (LOCK-DOWN BLOCK) Address = Block to lock/unlock/lock-down |
| WRITE (optional) | READ DEVICE ID | Data = 0x90 Address = Block address + offset 2 |

Table 61: Block Lock Operations Procedure (Continued)

| Bus Operation | Command | Comments |
|-----------------|-------------------|--|
| READ (optional) | BLOCK LOCK STATUS | Block lock status data Address = Block address + offset 2 |
| Idle | None | Confirm locking change on D[1, 0] |
| WRITE | READ ARRAY | Data = 0xFF Address = Block address |

Figure 35: Protection Register Programming Flow Chart

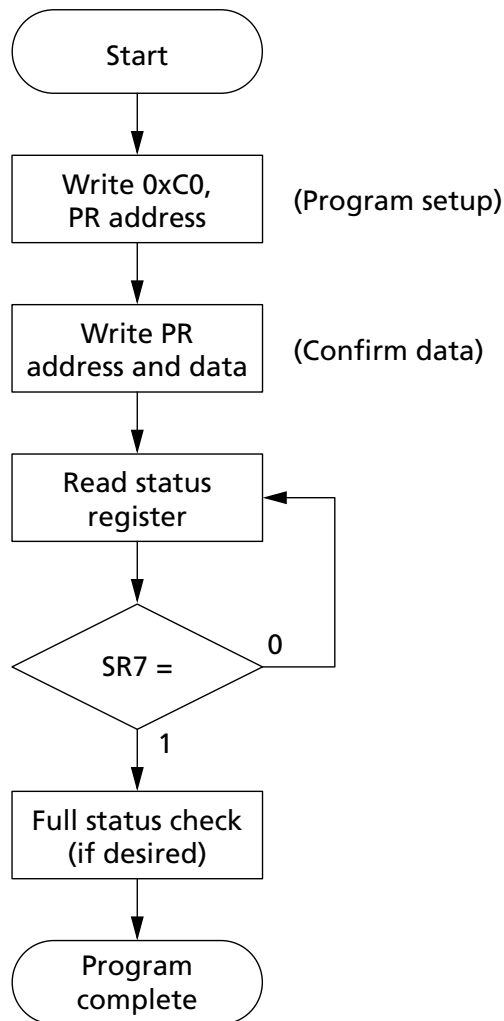


Table 62: Protection Register Programming Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|--------------------|--|-------|
| WRITE | PROGRAM PR SETUP | Data = 0xC0 Address = First location to program | 1, 2 |
| WRITE | PROTECTION PROGRAM | Data = Data to program Address = Location to program | 1 |
| READ | None | Status register data | 3 |
| Idle | None | Check SR7 1 = Write state machine ready 0 = Write state machine busy | 4 |

- Notes:
1. PROGRAM PROTECTION REGISTER operation addresses must be within the protection register address space. Addresses outside this space will return an error.
 2. Repeat for subsequent PROGRAM operations.
 3. Full status register check can be done after each PROGRAM operation or after a sequence of PROGRAM operations.
 4. Write 0xFF after the last operation to set to the read array state.

Figure 36: Protection Register Programming Full Status Check Flow Chart

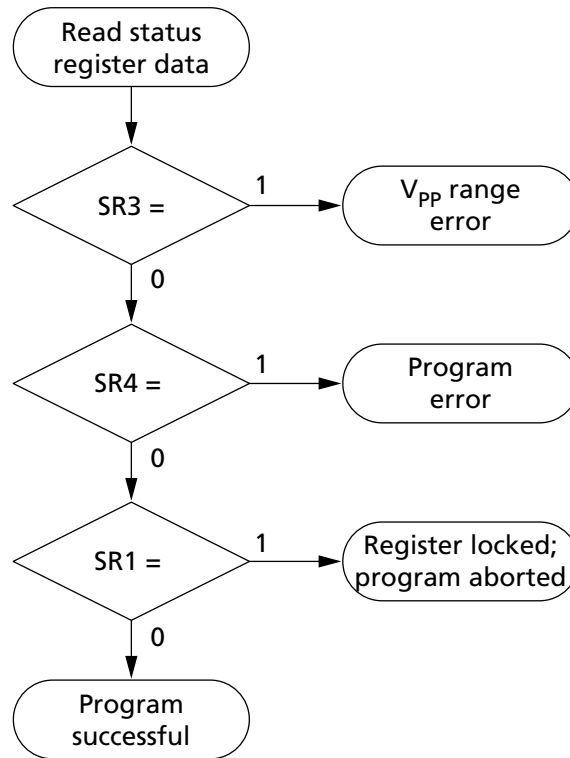


Table 63: Protection Register Programming Full Status Check Procedure

Notes apply to entire table.

| Bus Operation | Command | Comments | Notes |
|---------------|---------|---|-------|
| Idle | None | Check SR3 1 = V _{pp} error | – |
| Idle | None | Check SR4 1 = Programming error | – |
| Idle | None | Check SR1 1 = Register locked; operation aborted | – |

- Notes:
1. Only the CLEAR STATUS REGISTER command clears SR[1, 3, 4].
 2. If an error is detected, clear the status register before attempting a program retry or other error recovery.

Blank Check Flowcharts

Figure 37: Blank Check Flow Chart

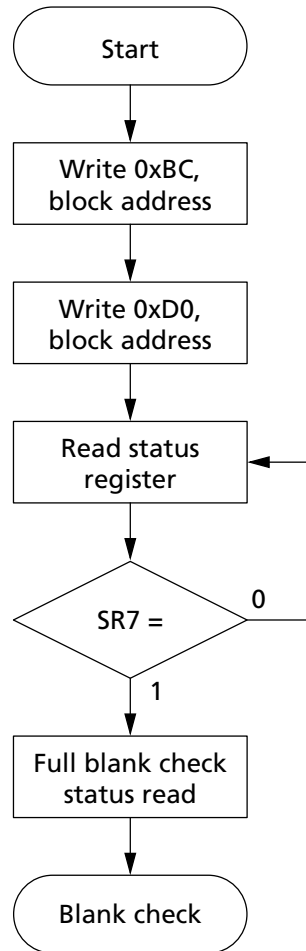


Table 64: Blank Check Procedure

| Bus Operation | Command | Comments | Notes |
|---------------|---------------------|--|-------|
| WRITE | BLANK CHECK SETUP | Data = 0xBC Address = Block to be read | 1 |
| WRITE | BLANK CHECK CONFIRM | Data = 0xD0 Address = Block to be read | - |
| READ | None | Status register data | 2 |
| Idle | None | Check SR7 1 = Write state machine ready 0 = Write state machine busy | - |

Notes: 1. Repeat for subsequent block blank check.



-
2. Full status register check should be read after blank check has been performed on each block.

Figure 38: Blank Check Full Status Check Flow Chart

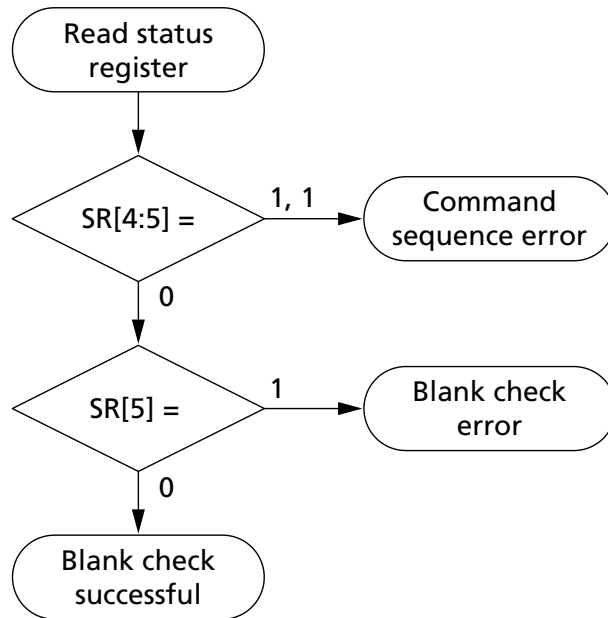


Table 65: Blank Check Full Status Check Procedure

Notes apply to entire table.

| Bus Operation | Command | Comments |
|---------------|---------|---|
| Idle | None | Check SR[4, 5] Both 1 = Command sequence error |
| Idle | None | Check SR5 1 = Blank check error |

- Notes:
1. SR[1, 3] must be cleared before the write state machine will allow blank check to be performed.
 2. Only the CLEAR STATUS REGISTER command clears SR[1, 3, 4, 5].
 3. If an error is detected, clear the status register before attempting a blank check retry or other error recovery.

AADM Mode

AADM Feature Overview

The following is a list of general requirements for AADM mode.

Feature availability. AADM mode is available in devices that are configured as A/D MUX. With this configuration, AADM mode is enabled by setting a specific volatile bit in the read configuration register.

High-address capture (A[MAX:16]). When AADM mode is enabled, A[MAX:16] and A[15:0] are captured from the A/ DQ[15:0] balls. The selection of A[MAX:16] or A[15:0] is determined by the state of the OE# input, as A[MAX:16] is captured when OE# is at V_{IL} .

READ and WRITE cycle support. In AADM mode, both asynchronous and synchronous cycles are supported.

Customer requirements. For AADM operation, the customer is required to ground A[MAX:16].

Other characteristics. For AADM, all other device characteristics (program time, erase time, I_{CCS} , etc.) are the same as A/D MUX unless otherwise stated.

AADM Mode Enable (RCR[4] = 1)

Setting RCR[4] to its non-default state (1b) enables AADM mode. The default device configuration upon reset or power-up is A/D MUX mode. Upon setting RCR[4] = 1, the upper addresses, A[MAX:16] are latched. All 0s are latched by default.

Bus Cycles and Address Capture

+ AADM bus operations have one or two address cycles. For two address cycles, the upper address (A[MAX:16]) must be issued first, followed by the lower address (A[15:0]). For bus operations with only one address cycle, only the lower address is issued. The upper address that applies is the one that was most recently latched on a previous bus cycle. For all READ cycles, sensing begins when the lower address is latched, regardless of whether there are one or two address cycles.

In bus cycles, the external signal that distinguishes the upper address from the lower address is OE#. When OE# is at V_{IH} , a lower address is captured; when OE# is at V_{IL} , an upper address is captured.

When the bus cycle has only one address cycle, the timing waveform is similar to A/D MUX mode. The lower address is latched when OE# is at V_{IH} , and data is subsequently outputted after the falling edge of OE#.

When the device initially enters AADM mode, the upper address is internally latched as all 0s.

WAIT Behavior

The WAIT behavior in AADM mode functions the same as the legacy non-MUX WAIT behavior (A/D MUX WAIT behavior is unique). In other words, WAIT will always be driven whenever DQ[15:0] is driven, and WAIT will tri-state whenever DQ[15:0] tri-state.

In asynchronous mode (RCR[15] = 1b), WAIT always indicates valid data when driven. In synchronous mode (RCR[15] = 0b), WAIT indicates valid data only after the latency count has lapsed and the data output data is truly valid.

Asynchronous READ and WRITE Cycles

For asynchronous READ and WRITE cycles, ADV# must be toggled HIGH-LOW-HIGH a minimum of one time and a maximum of two times during a bus cycle. If ADV# is toggled LOW twice during a bus cycle, OE# must be held LOW for the first ADV# rising edge and OE# must be held HIGH for the second ADV# rising edge. The first ADV# rising edge (with OE# LOW) captures A[MAX:16]. The second ADV# rising edge (with OE# HIGH) captures A[15:0]. Each bus cycle must toggle ADV# HIGH-LOW-HIGH at least one time in order to capture A[15:0]. For asynchronous reads, sensing begins when the lower address is latched.

During asynchronous cycles, it is optional to capture A[MAX:16]. If these addresses are not captured, then the previously captured A[MAX:16] contents will be used.

Asynchronous READ Cycles

For AADM, note that asynchronous read access is from the rising edge of ADV# rather than the falling edge (^tVHQV rather than ^tVLQV).

Table 66: AADM Asynchronous and Latching Timings

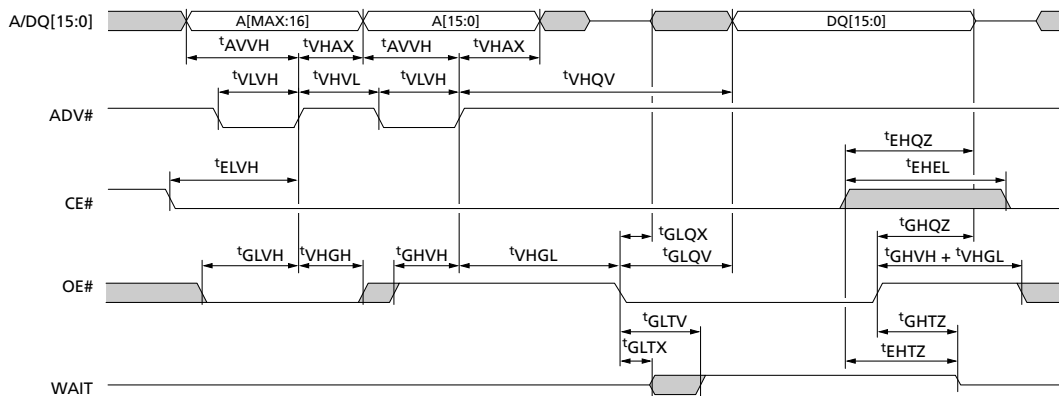
| Symbol | MIN (ns) | MAX (ns) | Notes |
|-------------------|----------|----------|-------|
| ^t GLQV | | 20 | |
| ^t PHQV | | 150 | |
| ^t ELQX | 0 | | |
| ^t GLQX | 0 | | |
| ^t EHQZ | | 9 | |
| ^t GHQZ | | 9 | |
| ^t OH | 0 | | |
| ^t EHEL | 7 | | |
| ^t ELTV | | 11 | |
| ^t EHTZ | | 9 | |
| ^t GLTV | | 7 | |
| ^t GLTX | 0 | | |
| ^t GHTZ | | 9 | |
| ^t AVVH | 5 | | |
| ^t ELVH | 9 | | |
| ^t VLVH | 7 | | |
| ^t VHVL | 7 | | |
| ^t VHAX | 5 | | |
| ^t VHGL | 3 | | |
| ^t VHQV | | 106 | 1, 2 |
| ^t PHVH | 30 | | |

Table 66: AADM Asynchronous and Latching Timings (Continued)

| Symbol | MIN (ns) | MAX (ns) | Notes |
|------------|----------|----------|-------|
| t_{GHVH} | 3 | | |
| t_{GLVH} | 3 | | |
| t_{VHGH} | 3 | | |

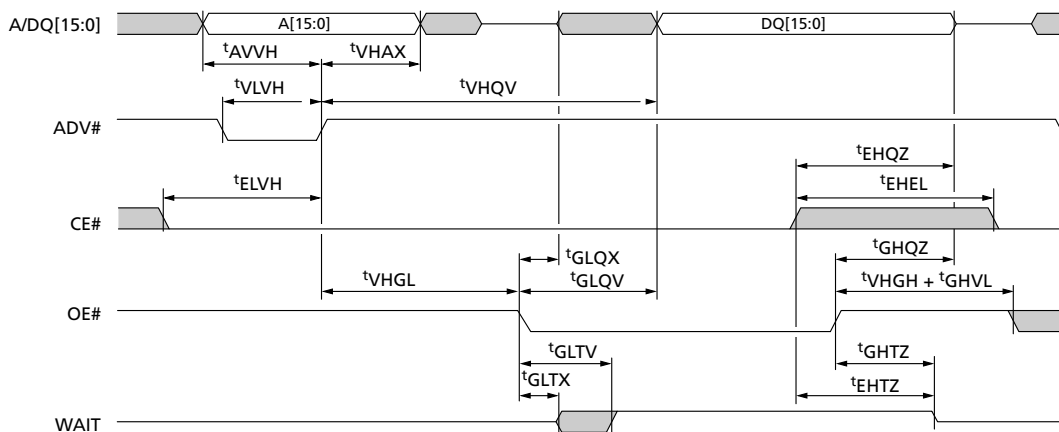
- Notes:
1. A READ cycle may be restarted prior to completing a pending READ operation, but this may occur only once before the sense operation is allowed to complete.
 2. t_{VHQV} applies to asynchronous read access time.

Figure 39: AADM Asynchronous READ Cycle (Latching A[$MAX:0$])



- Notes:
1. CE# need not be de-asserted at beginning of the cycle if OE# does not have output control.
 2. Diagram shows WAIT as active LOW (RCR[10] = 0).

Figure 40: AADM Asynchronous READ Cycle (Latching A[15:0] only)



- Notes:
1. Diagram shows WAIT as active LOW (RCR[10] = 0).
 2. Without latching A[$MAX:16$] in the asynchronous READ cycle, the previously latched A[$MAX:16$] applies.

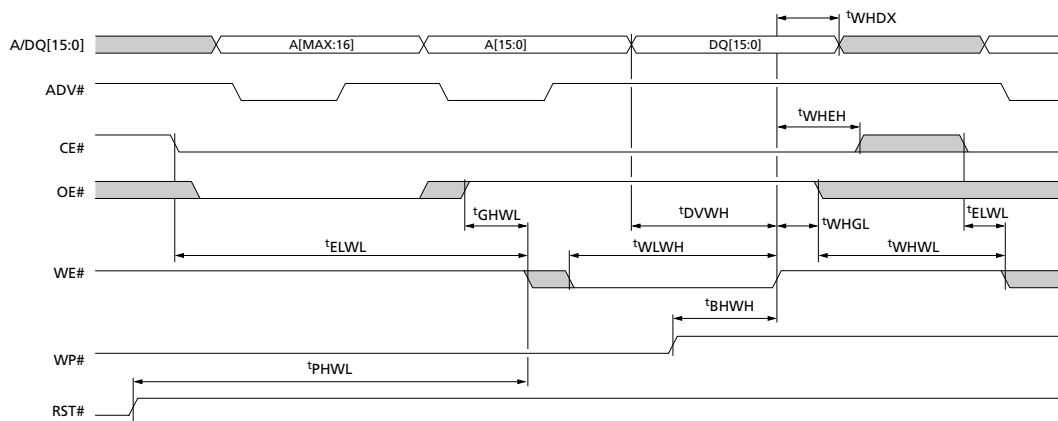
Asynchronous WRITE Cycles

Table 67: AADM Asynchronous Write Timings

| Symbol | MIN (ns) |
|------------|----------|
| t_{PHWL} | 150 |
| t_{ELWL} | 0 |
| t_{WLWH} | 40 |
| t_{DVWH} | 40 |
| t_{WHEH} | 0 |
| t_{WHDX} | 0 |
| t_{WHWL} | 20 |
| t_{VPWH} | 200 |
| t_{WVVL} | 0 |
| t_{BHWH} | 200 |
| t_{WHGL} | 0 |
| t_{GHWL} | 0 |

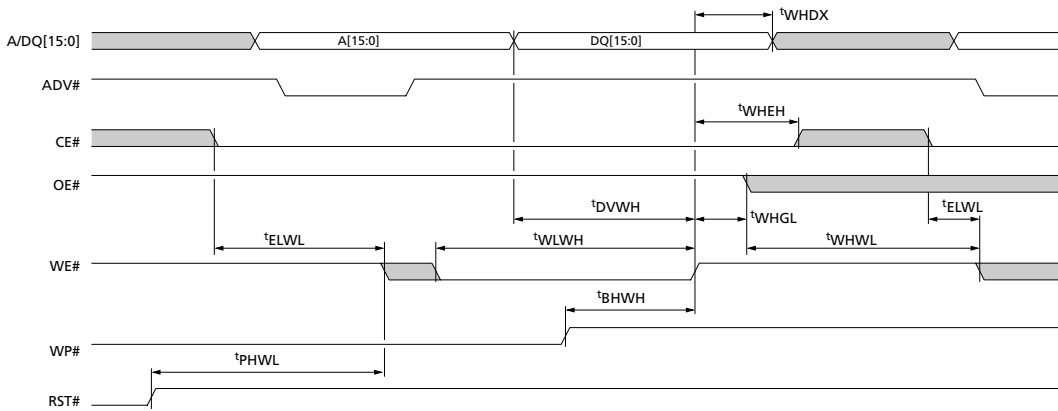
- Notes:
1. A READ cycle may be restarted prior to completing a pending READ operation, but this may occur only once before the sense operation is allowed to complete.
 2. t_{VHQP} applies to asynchronous read access time.

Figure 41: AADM Asynchronous WRITE Cycle (Latching A[**MAX:0**])



- Note:
1. CE# need not be de-asserted at beginning of cycle if OE# does not have output control.

Figure 42: AADM Asynchronous WRITE Cycle (Latching A[15:0] only)



Note: 1. Without latching A[MAX:16] in the WRITE cycle, the previously latched A[MAX:16] applies.

Synchronous READ and WRITE Cycles

Just as asynchronous bus cycles, synchronous bus cycles (RCR[15] = 0b) can have one or two address cycles. If there are two address cycles, the upper address must be latched first with OE# at V_{IL} followed by the lower address with OE# at V_{IH}. If there is only one address cycle, only the lower address will be latched and the previously latched upper address applies. For reads, sensing begins when the lower address is latched, but for synchronous reads, addresses are latched on a rising clock CLK instead of a rising ADV# edge.

For synchronous bus cycles with two address cycles, it is not necessary to de-assert ADV# between the two address cycles. This allows both the upper and lower address to be latched in only two clock periods.

Synchronous READ Cycles

For synchronous READ operation, the specifications in the AADM Asynchronous and Latching Timings Table also apply.

Table 68: AADM Synchronous Timings

Notes 1 and 2 apply to entire table

| Symbol | Target (104 MHz) Min (ns) | Target (104 MHz) Max (ns) | Notes |
|------------------------|------------------------------|------------------------------|-------|
| t _{CLK} | 9.6 | | |
| t _{RISE/FALL} | | 1.5 | 3 |
| t _{AVCH} | 3 | | |
| t _{VLCH} | 3 | | |
| t _{ELCH} | 3.5 | | |
| t _{CHQV} | | 7 | |
| t _{CHQX} | 1 | | |
| t _{CHAX} | 5 | | 4 |

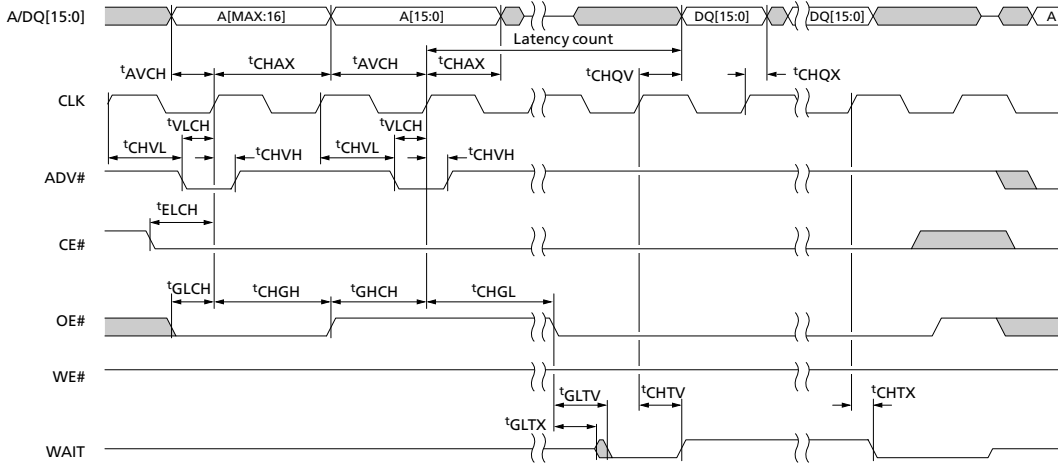
Table 68: AADM Synchronous Timings (Continued)

Notes 1 and 2 apply to entire table

| Symbol | Target (104 MHz) Min (ns) | Target (104 MHz) Max (ns) | Notes |
|-------------------|------------------------------|------------------------------|-------|
| ^t CHTV | | 7 | |
| ^t CHVL | 2.5 | | |
| ^t CHTX | 1 | | |
| ^t CHVH | 2 | | |
| ^t CHGL | 2.5 | | 6, 4 |
| ^t VLVH | ^t CLK | 2 x ^t CLK | 5, 6 |
| ^t VHCH | 3 | | |
| ^t CHGH | 2 | | |
| ^t GHCH | 2 | | |
| ^t GLCH | 3 | | |

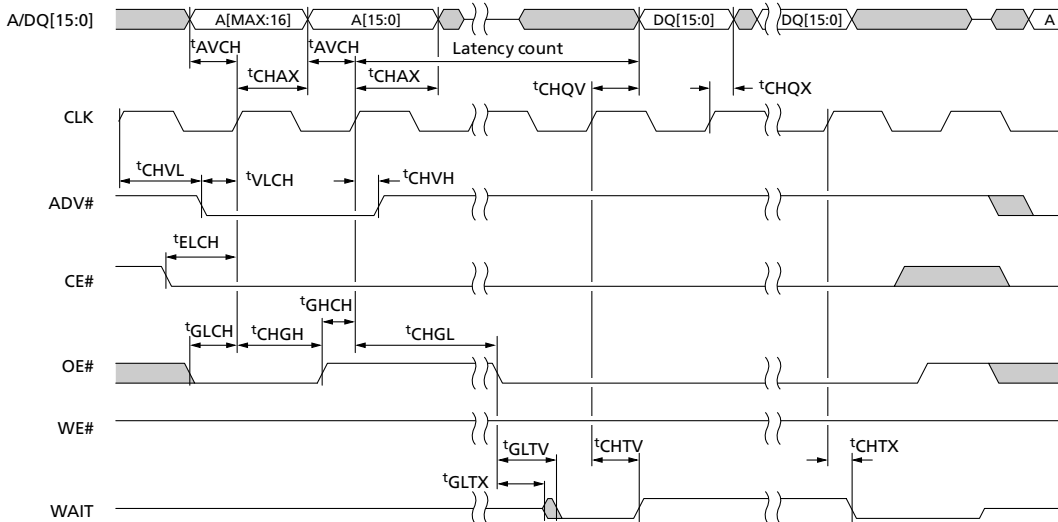
- Notes:
1. In synchronous burst READ cycles, the asynchronous OE# to ADV# setup and hold times must also be met (^tGHVH and ^tVHGL) to signify that the address capture phase of the bus cycle is complete.
 2. A READ cycle may only be terminated (prior to the completion of sensing data) one time before a full bus cycle must be allowed to complete.
 3. Rise and fall time specified between V_{IL} and V_{IH}.
 4. To prevent A/D bus contention between the host and the memory device, OE# may only be asserted LOW after the host has satisfied the ADDR hold spec, ^tCHAX.
 5. The device must operate down to 9.6 MHz in synchronous burst mode.
 6. During the address capture phase of a read burst bus cycle, OE# timings relative to CLK shall be identical to those of ADV# relative to CLK.

Figure 43: AADM Synchronous Burst READ Cycle (ADV# De-asserted Between Address Cycles)



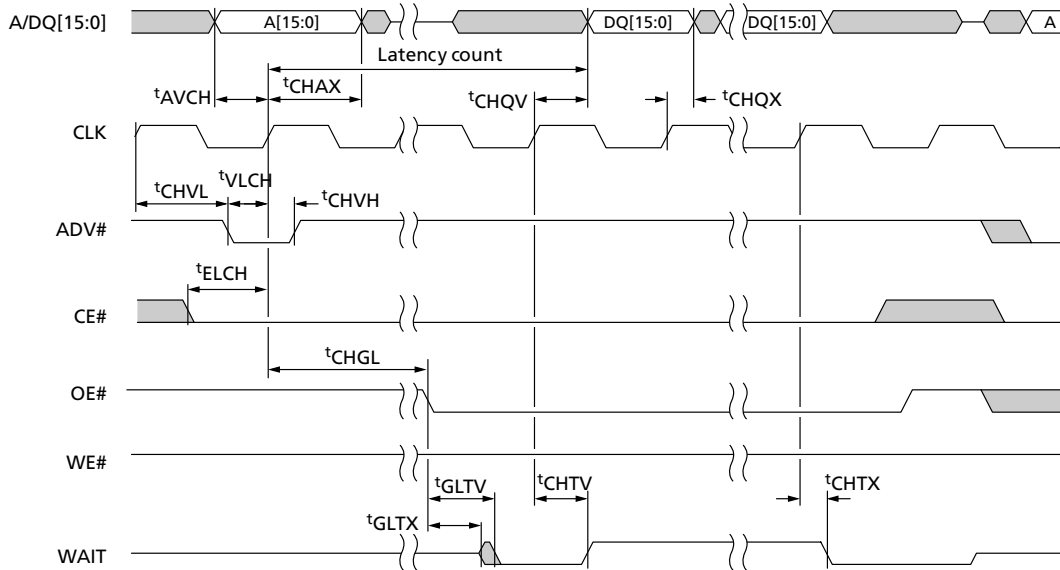
- Notes:
1. CE# need not be de-asserted at beginning of cycle if OE# does not have output control.
 2. Diagram shows WAIT as active LOW (RCR[10] = 0) and asserted with data (RCR[8] = 0).
 3. For no-wrap bursts, end-of-wordline WAIT states could occur (not shown).

Figure 44: AADM Synchronous Burst READ Cycle (ADV# Not De-asserted Between Address Cycles)



- Notes:
1. CE# need not be de-asserted at beginning of cycle if OE# does not have output control.
 2. Diagram shows WAIT as active LOW (RCR[10] = 0) and asserted with data (RCR[8] = 0).
 3. For no-wrap bursts, end-of-wordline WAIT states could occur (not shown).

Figure 45: AADM Synchronous Burst READ Cycle (Latching A[15:0] only)



- Notes:
1. Diagram shows WAIT as active LOW (RCR[10] = 0) and asserted with data (RCR[8] = 0).
 2. For no-wrap bursts, end-of-wordline WAIT states could occur (not shown).
 3. Without latching A[MAX:16] in the synchronous READ cycle, the previously latched A[MAX:16] applies.

Synchronous WRITE Cycles

For synchronous writes, only the address latching cycle(s) are synchronous. Synchronous address latching is depicted in the Synchronous READ Cycles.

The actual WRITE operation (rising WE# edge) is asynchronous and is independent of CLK. Asynchronous writes are depicted in Asynchronous WRITE Cycles.

System Boot

Systems that use the AADM mode will boot from the bottom 128KB of device memory because A[MAX:16] are expected to be grounded in-system. The 128KB boot region is sufficient to perform required boot activities before setting RCR[4] to enable AADM mode.

Revision History

Rev. F – 5/18

- Added Important Notes and Warnings section for further clarification aligning to industry standards

Rev. E – 6/15

- Updated endurance and cycle limits in Features
- Updated frequencies in the Supported Latency and Clock Frequency table in Read Configuration Register

Rev. D – 4/15

- Updated DC Voltage Characteristics and Operating Conditions table in Electrical Specifications

Rev. C – 2/15

- Revised initial access speed specification: 104ns to 106ns.
- Revised clock frequency specifications.
- Revised V_{CC} program and erase and AC read specifications.
- Revised AADM asynchronous and latching timings.

Rev. B – 10/14

- Minor format edit: moved Supported Latency and Clock Frequency table to the end of the read configuration register topic.
- Revised t_{CLK} from 9ns to 9.6ns in the AADM Synchronous Timings table.

Rev. A – 12/13

- Initial release

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-4000
www.micron.com/products/support Sales inquiries: 800-932-4992
Micron and the Micron logo are trademarks of Micron Technology, Inc.
All other trademarks are the property of their respective owners.

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.